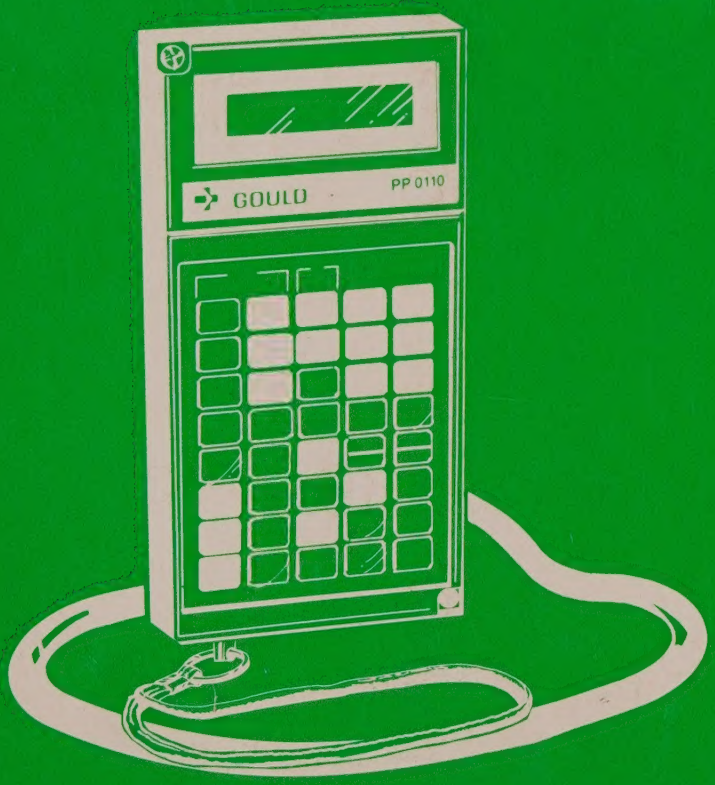
GM-WCPC-001 REV. A

Gould
PP 0110 Programming Guide



Bentley's - 505.344.1812
07002167
901NM ~ 901NM ~ 901NM
www.bentleysauction.com

The Industrial Automation Systems Group, (IASG), of Gould Inc., as a result of its sale to AEG of West Germany, has changed its name to:
MODICON.
For further details, contact:
Marketing Communications, Dept. A, MS 3-4B
One High Street, North Andover, MA 01845

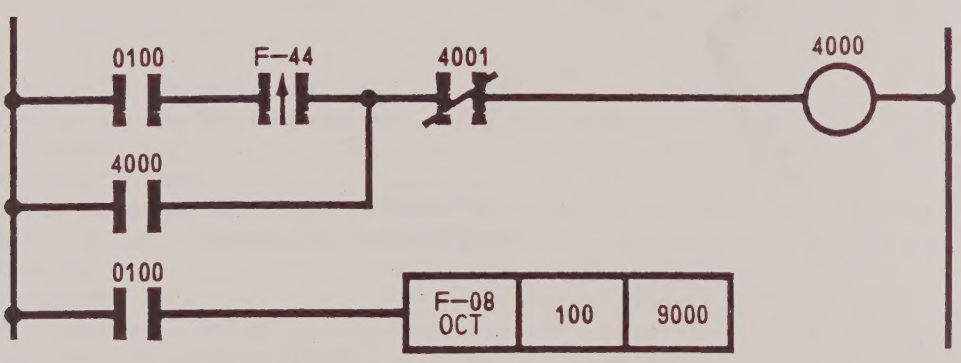
GOULD
Electronics

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_11361

GOULD PP 0110

PROGRAMMING GUIDE



The information in this document is subject to change without notice and should not be construed as a commitment by Gould Inc., Programmable Control Division. Gould, Inc. assumes no responsibility for errors that may appear in this document. No part of this document may be reproduced in any form without the express written permission of Gould, Inc., Programmable Control Division. All rights reserved.

The following are trademarks of Gould, Inc.:

PP 0110 PC 0185 PP 0105

TABLE OF CONTENTS

Chapter 1 - Programmer Features and Specifications

Parts and Functions....1-1
Dimensions and Specifications....1-3
Keyboard....1-5
Buzzer....1-6
EL Illumination....1-6
LCD Display....1-7
Installation
  Direct Installation on Controller....1-8
  Installation on Control Panel....1-9
  Hand Held or Desktop Operation....1-10
  Hints for Connection....1-12
  Liquid Crystal Contrast Adjustment....1-12
Programmer Modes of Operation
  Mode Setup....1-14
  Mode Functions....1-14

Chapter 2 - Programming

Memory Set Up and Clear....2-1
System Design Considerations....2-3
I/O Numbering....2-4
Special Relays....2-5
Relay Byte and Absolute Addressing....2-6
Basic Relay Programming Instructions....2-8
Timers....2-13
Counters....2-15
Maintenance Display....2-17
Application Instructions....2-23

Chapter 3 - PROGRAM, MONITOR, and CHANGE Mode Commands

Program Entry Sequence....3-1
Addressing Procedure....3-2
PROGRAM Mode
  Address Setup....3-4
  Memory Clear....3-5
  System Memory Read/Write....3-8
  Program Entry and Verification....3-10
  Program Check....3-15
  Program Search....3-21
  Program Correction....3-24
  Recording and Verification....3-28
  Saving the Program....3-32
  Loading the Program....3-34

TABLE OF CONTENTS

Chapter 3 - PROGRAM, MONITOR, and CHANGE Mode Commands (Cont)

MONITOR Mode
Program Monitor....3-35
Data Memory Monitor....3-42
CHANGE Mode
Monitor Functions....3-45
Setting Latched Relays....3-45
Setting TMR, CNT, and MD....3-46
Setting Register Values....3-47
Setting TMR, CNT, and MD Preset Values....3-48

Chapter 4 - Memory Maps

Data Memory....4-1
System Memory....4-3
Program Memory....4-7

Chapter 5 - Programming Notes

Programmer Warnings....5-1
Number Systems....5-2
Source and Destination....5-7
Application Instruction and Stack Register....5-8
Operation Conditionals....5-13
Flags....5-15
Double Length Operations....5-19
Using the Relay Area....5-24
Data Memory Block and Reference Addresses....5-25
Signal Inputs and Outputs....5-30

Appendix A - Key Functions and Operations....A-1

Appendix B - Display Message List....A-7

Appendix C - Data Memory....A-11

Appendix D - Bit Manipulation for Basic Instruction Set....A-15

Parts and Functions

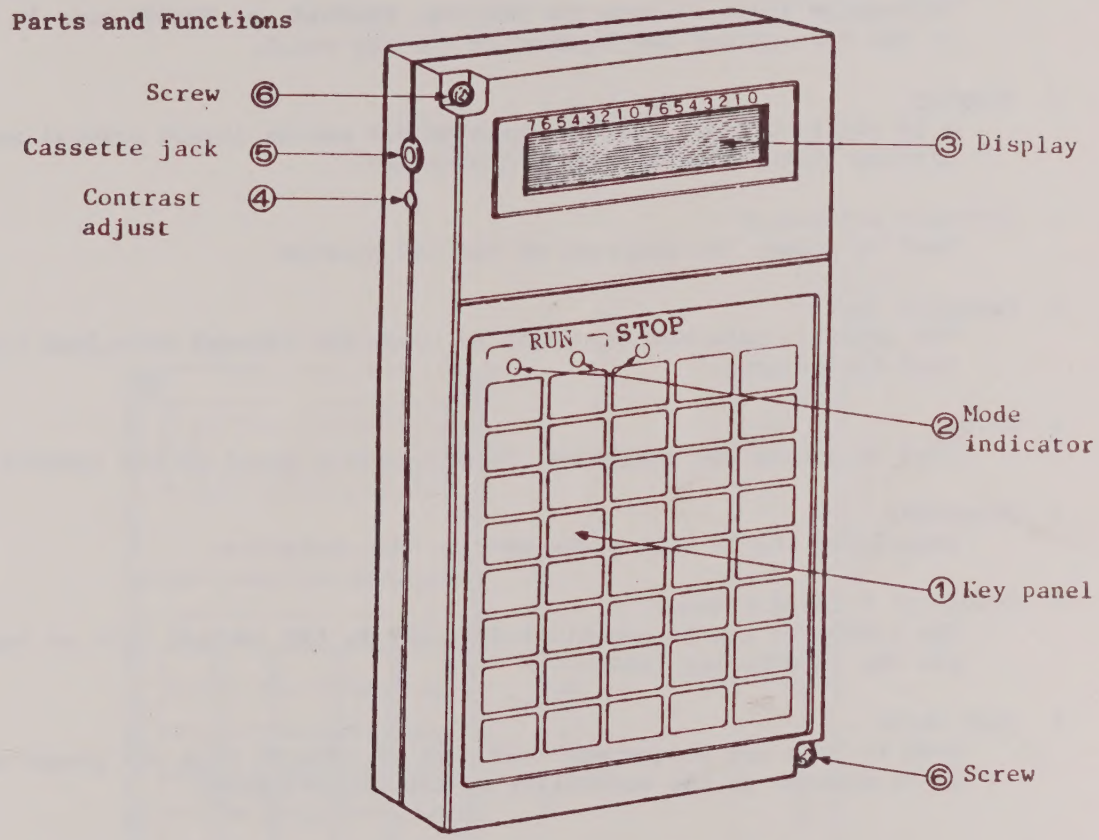


Fig. 1, PP 0110 Programmer (front)

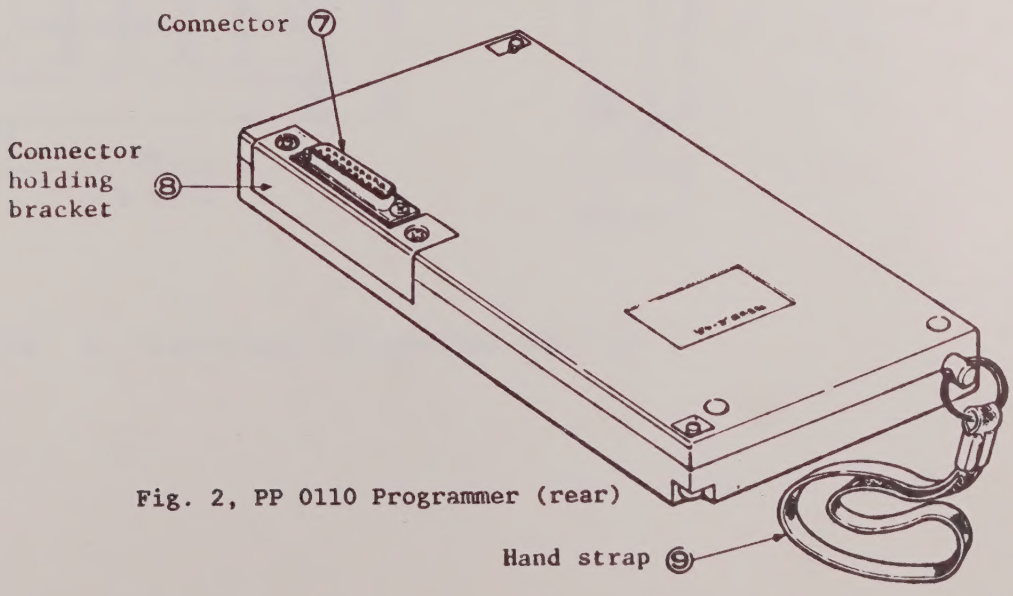


Fig. 2, PP 0110 Programmer (rear)

1 Key Panel
Used to enter a change or to monitor a program.

2 Mode Indicators
Indicators light up when the MONITOR, PROGRAM, or CHANGE mode is selected by the mode select key located on the key panel.

3 Display
A 16 digit, 2 line, back illuminated dot matrix liquid crystal is used to display instruction and data information.

4 Contrast Adjustment
Used to adjust the contrast of the LCD display.

5 Cassette Jack
The cassette tape recorder must be connected through this jack to save or load the program.

6 Screw
Used to secure the programmer to the control panel or the controller.

7 Connector
Interfaces the PP 0110 programmer to the controller.

8 Connector Holding Bracket
The connector can be installed directly to the control unit or remotely via the interfacing cable.

9 Hand Strap
Used to hold the programmer. It must be removed from the programmer when it is mounted on the controller or the control panel.

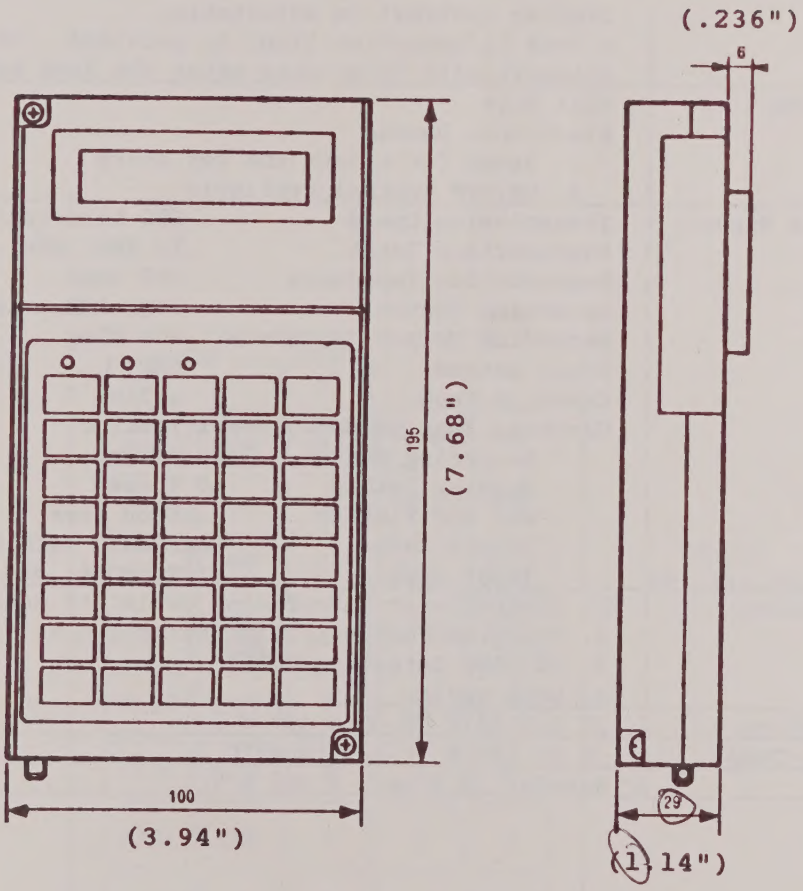
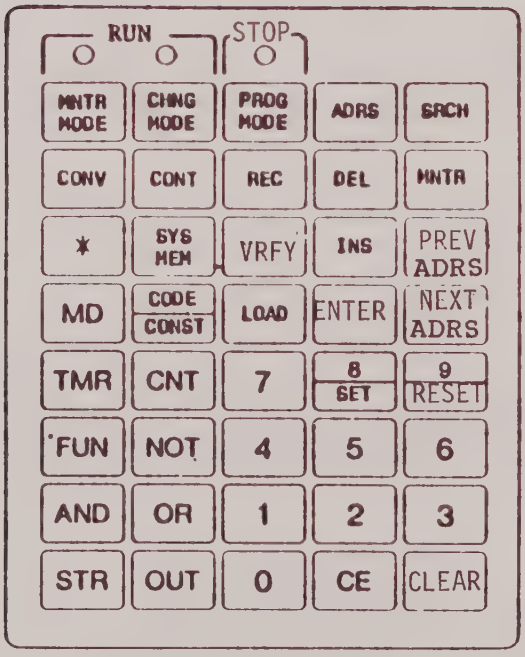


Fig. 3, Programmer Dimensions MM (ins.)

PERFORMANCE SPECIFICATIONS

ITEM	SPECIFICATION
Connection with control unit, PC 0185	Connection method (connector location variable) 1. Direct Installation 2. Connection with the 3 meter interfacing cable. Signal level: RS-422 compliance (19.2K baud)
Display Elements	16 digit, 2 line, dot matrix LCD 1. Capital and small alphabetics 2. Numerics 3. Symbols Display contrast is adjustable. A back illumination light is provided. It will turn off automatically 10 minutes after the last key entry.
Keyboard	Flat Keys Electronic Buzzer 1. Beeps for errors and key entry 2. ON/OFF control available
Cassette Signal	Transmission Speed 600 bits/sec Reproduction Input 1V RMS, min. Reproduction Impedance 200 Ohms Recording Output 10mV RMS, min. Recording Output Impedance 200 Ohms Check Method CRC Cassette Tape audio Cassette Tape Recorder Specification: Recording Method AC Biased Erasing Method AC Erased Wow and Flutter 0.2% or less Output Jack EARPHONE JIS C 6560, 3.5 Input Jack MICROPHONE JIS C 6560, 3.5
Accessories	1. Control unit interfacing cable (3 meters) 2. Cassette recorder interfacing cable (1.5 meters) 3. PC 0085 Interface Cable 4. Lock spring
Temperature	32 to 125°F (0 to 40° C)
Storage Temp	-4 to 140°F (-20 to 60°C)
Weight	Approx. 12.5 oz. (350 g)

Keyboard



Model
PP 0110-000

NOTE
The remainer of this document is written and diagrammed to show the PP 0110-001 model. Refer back to this page for keypad changes.

Fig. 4, Programmer Keyboards

Buzzer ON/OFF

The following buzzers are provided for the programmer.

1. Key Touch Tone - Beeps once for each key press.
2. Error Alert Tone - Beeps twice for an operational error.

(How to make ON/OFF choice - default state is ON)

EL Back Illumination

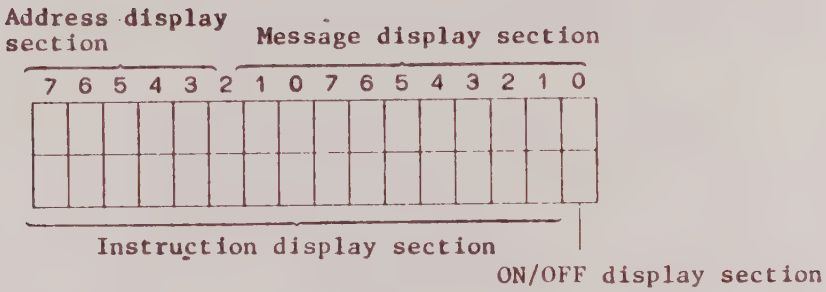
The LCD depends on an external light source to illuminate properly. It may be necessary under low light conditions to use this feature to supply extra light.

(How to make EL Illumination ON/OFF choice - default state is OFF)

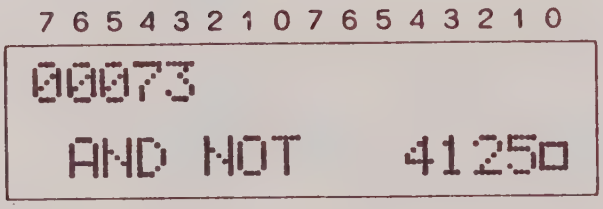
An automatic power off feature is provided to prolong the life of the lighting element. The back illumination will turn OFF 10 minutes after the last key entry. It can be turned ON by using the steps shown above.

Display

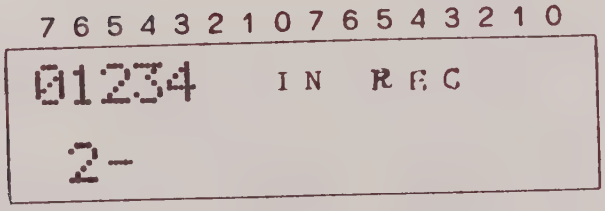
A 16 digit, 2 line, 5x7 dot matrix liquid crystal is used for the display.


Address display section
Message display section
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
Instruction display section
ON/OFF display section

Example 1



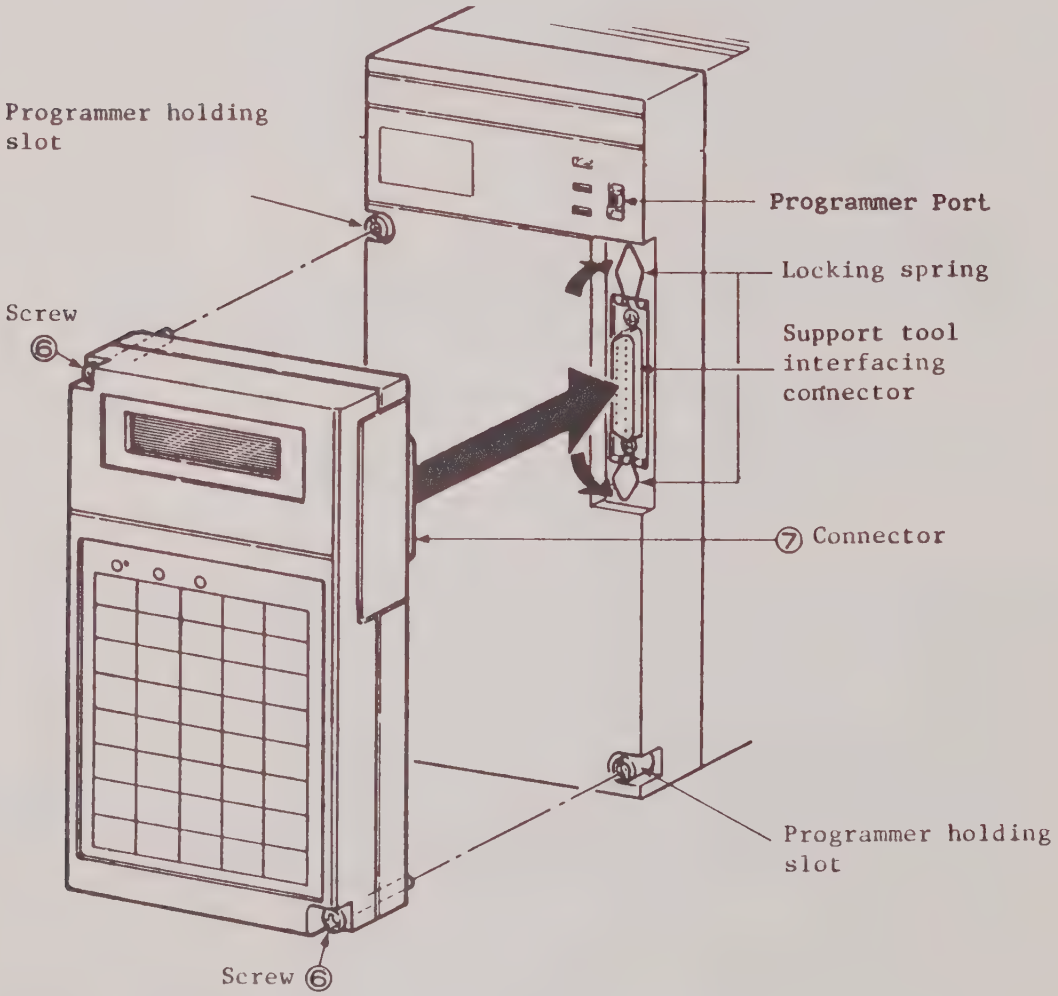
Example 2



Installation

Direct Installation On the Control Unit

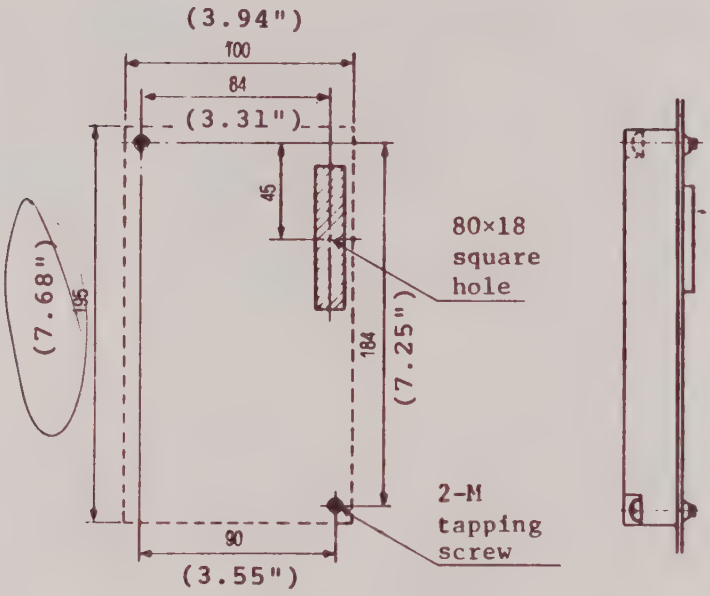
The following describes the installation procedure for connection to the PC 0185 Controller. The PC 0185-001 model has a 25 pin connector, the PC 0185-000 model has a 15 pin connector.



1. Set the Programmer Port switch of the controller to OFF.
2. Turn the programmer connector lock spring towards the controller side.
3. Fasten connector 7 from the programmer to the connector on the controller.
4. Fasten programmer securing screw 6 to the programmer holding slot of the controller.
5. Set the Programmer Port switch of the controller to ON.

Installation On a Control Panel

It is possible to install the controller and base unit in a control panel. The programmer can then be installed on the panel cover.



Dimensions in MM (inches)

1. Install the lock spring on the connector of the programmer in order to secure the interfacing cable.

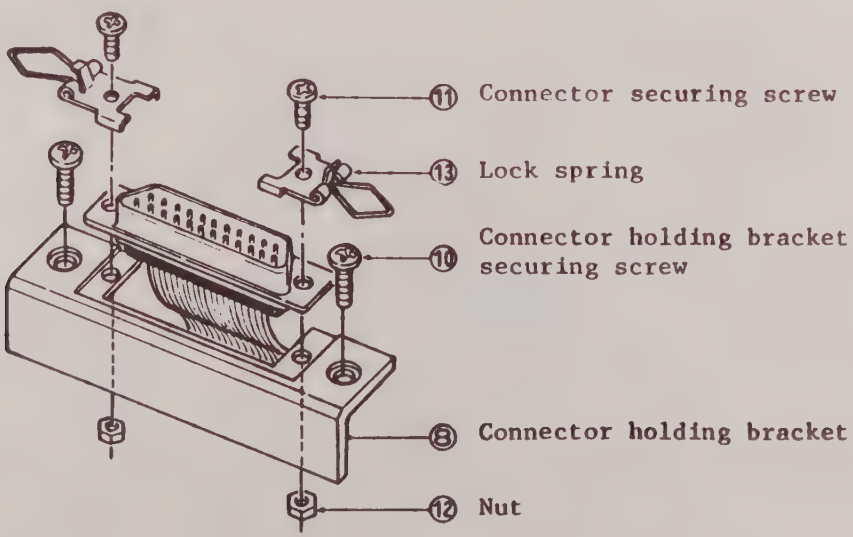


Fig. 5, Connector Installation

1. Remove two screws (10) that hold the connector holding bracket (8).
2. Remove two connector fastening screws (11).
3. Install two lock springs (13) as shown in Fig. 7.
4. Replace and secure the connector holding bracket.

2. Install the programmer on the control panel cover through the hole in the cover.

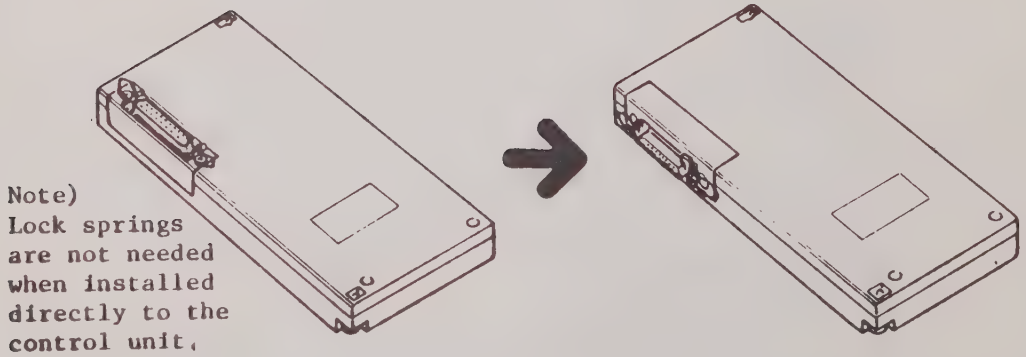
3. Connect the programmer to the controller with the interfacing cable. Fasten all connectors and lock springs.

Note 1: Keep Programmer Port switch in OFF until the connection is complete.
Note 2: Keep the interfacing cable away from high voltage lines, power supply lines, and I/O signal lines.

Hand Held and Desktop Operation

The programmer can be hand held or mounted on a desktop with the use of the interfacing cable.

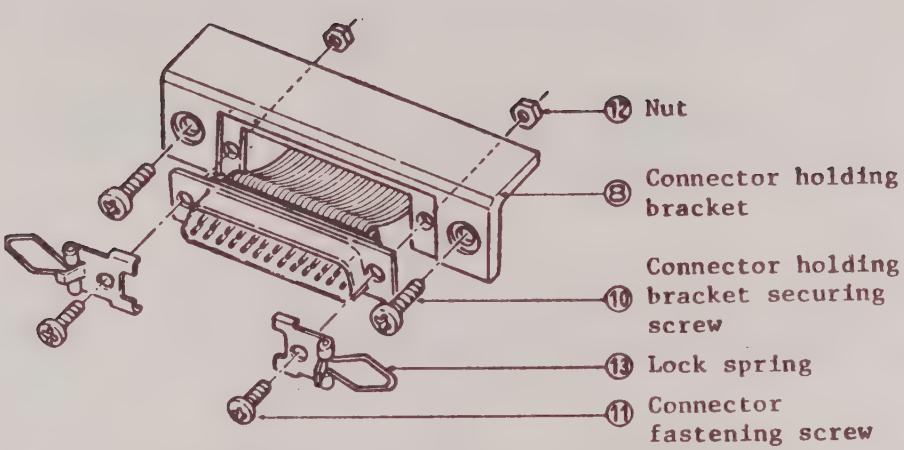
1. The connector holding bracket must face the side instead of the back.



Direct installation to the controller or control panel

Hand held or desktop operation

Fig. 6, Connector Location for Direct and Hand Held Operation

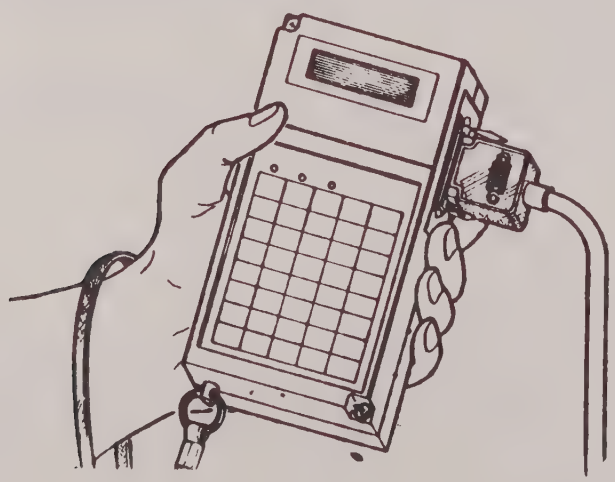


1. Remove two screws (10) that hold the connector holding bracket (8).
2. Remove two connector fastening screws (11).
3. Install two lock springs (13).
4. Secure the connector holding bracket with screw (1) so that the connector is on the side of the programmer.

2. Connect the programmer to the controller with the interfacing cable. Fasten all connectors and lock springs.

Note 1: Set the Programmer Port switch in OFF until the connection is complete. It should be placed in the OFF position any time the programmer is removed or installed.

Note 2: It is recommended that the strap be used when the programmer is hand held.



Hints for Connection

Observe the following cautions when connecting the programmer with the control unit.

1. Set the Programmer Port switch to OFF until the connection is complete. It should be placed in the OFF position any time the programmer is removed or installed.

ON = Programmer communication enabled

OFF = Programmer communication disabled

An error may be indicated if the programmer is connected or disconnected with the switch set in the ON position.

2. If the operation is to be done without the aid of the attached programmer, the switch has to be set to the OFF position.

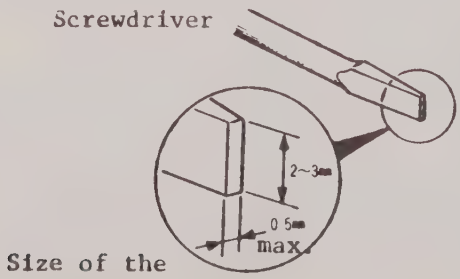
3. The following message will be displayed when the switch is set to the OFF side.

SET CU SW TO IN-
USE TO USE PG

4. When the switch is set to the OFF position or when the programmer is detached, the controller assumes the operational mode that is set by the program when the switch is at the ON position. When the switch is again set to ON, the controller assumes the operational mode that is set in the programmer.

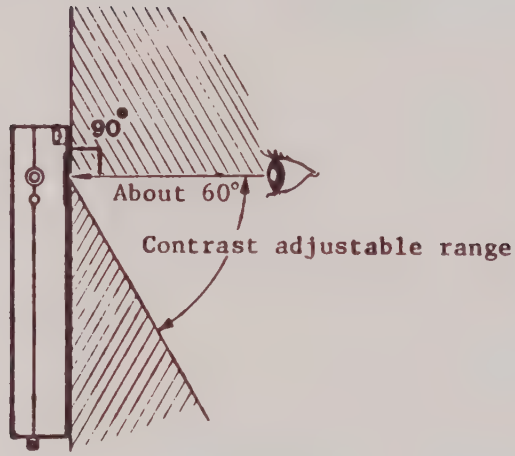
Note 1: When the programmer is connected to the controller with the interfacing cable, keep the cable away from high voltage lines, power supply lines, and I/O signal lines.

Liquid Crystal Contrast Adjustment



Size of the screwdriver tip

The contrast of the LCD will change depending on viewing angle. To adjust for best contrast, turn the screw located on the side of the programmer.



Note 1: The response time and contrast of the LCD will vary depending on the temperature.

Programmer Modes of Operation

There are three modes of operation: PROGRAM, MONITOR, and CHANGE.

1. Mode Setup
   Use the following key strokes to enter the desired operational mode. The respective LED indicator on the keyboard will light to indicate the current mode of operation.

PROGRAM	0 0 PROG MODE 0 SET
MONITOR	0 0 MNTR MODE 0 SET
CHANGE	0 0 CHNG MODE 0 SET

2. Functions of Modes

Mode	Controller State	Controller RUN Indicator	Functions
PROGRAM	Stop	Blinking	1. Specification setup (write to system memory) 1. Change latched relay area. 2. Choice of TMR reset at power interrupt 3. Choice of CNT reset 4. Choice of output reset at PC failure 5. Set memory capacity 6. Set file memory size 7. Registration of number during link 2. Memory Clear 3. Enter, insert, or delete an instruction 4. Search data memory for instruction 5. Sequential read of instructions 6. Program check 7. Parity check 8. Tape recorder operations (load, record, verify)

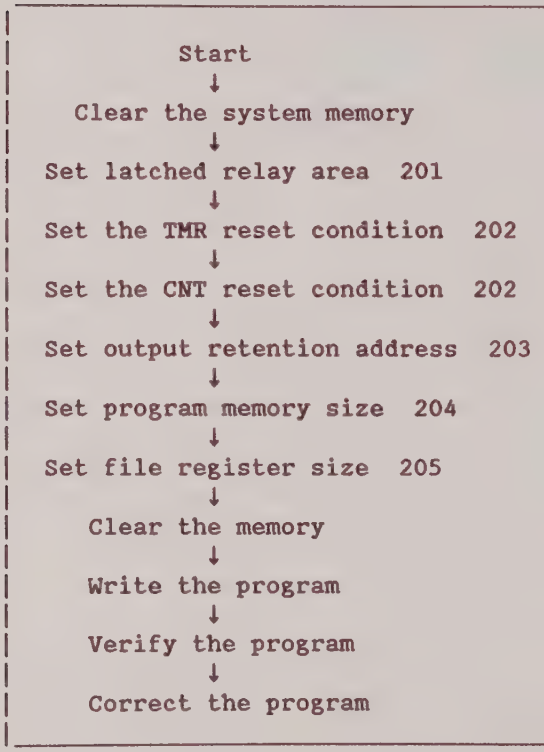
Functions of Modes (Cont.)

Mode	Controller State	Controller RUN Indicator	Functions
MONITOR	Run	Steady ON	1. Read of an instruction and continuity 2. Read relay's ON/OFF state 3. Read current TMR, CNT, and MD value 4. Read in current register value 5. Tape recorder operation and data memory 6. Search of an instruction and data memory
CHANGE			7. Set/reset latch relays 8. Set/reset TMR and CNT 9. Change preset values 10. Change constant 11. Change register value

NOTE: Items 1 through 6 above are applicable only in the MONITOR mode. Items 1 through 11 are applicable in the CHANGE mode.

System Memory Setup and Clear

After the completion of the ladder design, the program must be loaded in reference to the procedure shown below:


Start
Clear the system memory
Set latched relay area 201
Set the TMR reset condition 202
Set the CNT reset condition 202
Set output retention address 203
Set program memory size 204
Set file register size 205
Clear the memory
Write the program
Verify the program
Correct the program

When the system memory is cleared all the setups below return to their initial state. If they are used in their initial states, no setup is required.

System Memory Address (octal)	Contents after clear (octal)	Initial State
200	300	Latched relay 7000 ~ 7337
201	000	TMR reset
202		Reset when CNT turns ON
203		All outputs retained at HALT
204	001	Program memory size
205		not used

Note 1: With memory clear the range of the area to be cleared is judged by the state of 204 and 205 in the following process:

The entire program memory area is padded with NOP instructions and the END instruction is written into the last address.

Data memory and file registers are all set to zero. Therefore, the memory must be cleared after setup of the system memory.

System Memory Clear Procedure

The following procedure must be used to set the system memory to its initial state.

Key operation	Display
[ ] [ ] [PROG MODE] [8 SET] [CLR] [9 RST]	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 OOOOO MEM CL
[SYS MEM]	SYS MEM CL
[ENT]	SYS MEM CL OK

Fig. 7, System Memory Clear

System Design Considerations and Safety

The main difference between a PC and a conventional relay circuit is that the PC handles the control program in a cyclic manner (serially), while the relay circuit processes in parallel.

The PC's programming elements should not include any critical components. Your application must provide fail-safe mechanisms that are independent of the PC's program.

The circuits described below should be established outside the PC:

- Emergency stop circuit
- Protective circuits
- High voltage operating circuits
- Other circuits that prevent human injury and/or
- mechanical failures

The response time of the PC must be considered when making fail safe plans.

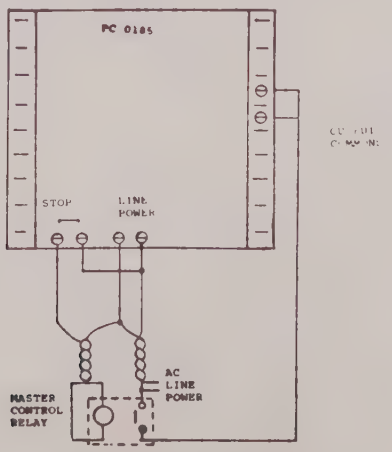
CAUTION

The PC 0185 Controller is configured in the factory with all outputs in the retentive state. This means that if an output is ON when a logic scan stops (changing from RUN mode to STOP mode via the programmer, or if there is a diagnostic failure), it will remain ON. In some applications this may be undesirable or dangerous. The control system designer is responsible for PC operation and safety. There are two methods of making all outputs non-retentive.

1. To turn all of the outputs OFF when the scan stops set system address #203 to 377.
Procedure: Press keys as in the sequence shown

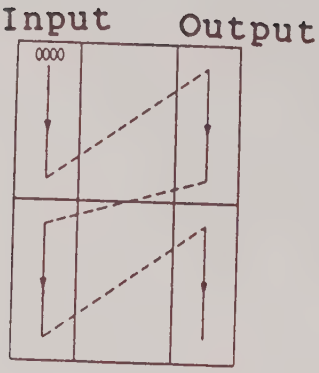
*, *, PROG MODE, 8/SET, CLEAR, SYS MEM, 2, 0, 3, MON, 3, 7, 7, ENTER

2. To turn all of the outputs OFF when the scan stops, power all of the output field wiring from a master control relay contact. The master relay is controller by the HALT output terminals on the PC 0185 terminal connector.

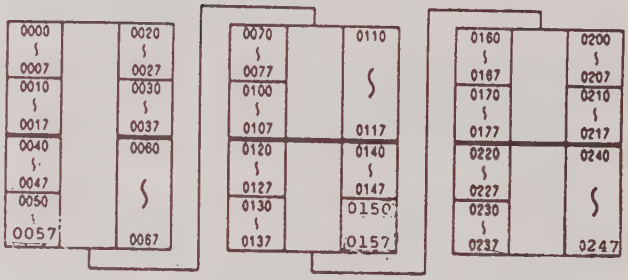


I/O NUMBERING

The relay numbers of the I/O units must be assigned a sequential number from the input side to the output side.

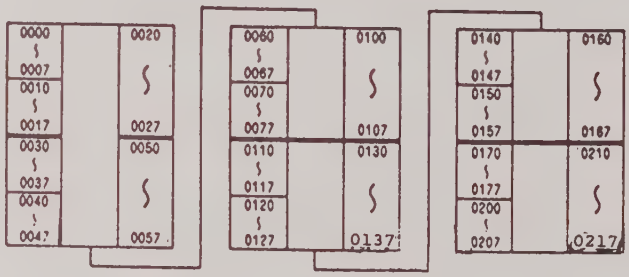


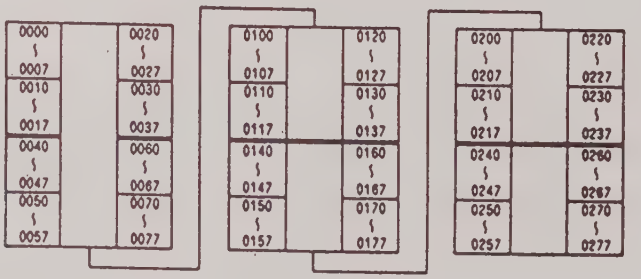
(1) When only 32-point I/O units are used their locations vs. relay numbers will be as shown on opposite:



(2) When only 24-point I/O units are used, their locations vs. relay numbers will be as shown on opposite:

(3) When 32-point I/O units are used with 24-point I/O units, for example, their locations vs. relay numbers will be as shown on opposite:





Special Relay Area

1. 7340 ~ 7347 (].734)

The special area that stores the error will store the error with the highest priority, if more than one error occurs.
The error code will be removed as soon as the cause is eliminated.

2. 7354 ~ 7357 (flag)

The relevent flag will be automatically set according to the operation.

3. 7360 and 7364 (clocks)

Used for the clock of the CNT instruction and other application instructions.

4. 7365 (preset value change coil)

It is possible in the preset change mode to set and reset latched relays (7000~7337). But, 7365 should be set ON in the program to set or reset relays (0000~4777); or to write data from the programmer into registers (].000~].733, b.000~b.377, 9000~9377,) or the file registers (30000~33777).

Programming Ever-ON

The Ever-OFF contact 7366 is used

NOTE
7365 MUST be programmed ON to use the 9000 series registers.

Program to set ON by external switch

With switch connected to 0010 ON, 7365 is ON

5. 7366 (Ever-OFF contact)

Used to program an Ever-OFF contact (type-a) or Ever-ON (type-b).

6. 7367 (zero cross switch)

Set ON when not required

7. 7370 ~ 7377 (self diagnostic result)

If an error is detected after self-diagnostics, the relevent contact will be set ON.

7370	Parity Error	7374	Option Failure
7371	CPU Failure	7375	-
7372	Battery Failure	7376	ROM Failure
7373	I/O Failure	7377	Power Supply Failure

TMR, CNT, MD Data Storage Area

The 256 bytes of b.000 to b.377 is the area in which the CNT current value and MD information are stored. Since the total of TMR, CNT, and MD is 128 points, two bytes are used for each point. See the table for the relationship of the TMR, CNT, and MD numbers to the b.xxx area.

The TMR or CNT current value can be used for operation when b.000 ~ b.377 is specified by an application instruction like F-00.

Relay Area Byte Address and Absolute Address

The PC 0185 controller not only carries out bit term operations like AND and OR, it also performs four math functions and data transfer. A data processing instruction is handled in byte terms.

In order to permit data processing the I/O relay, link relay, internal relay, and latched relay areas are also byte term. The PC 0185 has two ways of accessing the byte term address; byte address and absolute address.

A. Byte Address

The byte term address corresponds to the relay number. The least significant digit of the four digit relay number is discarded and the symbol]. is prefixed to the three remaining digits.

0137	0136	0135	0134	0133	0132	0131	0130

The byte address is].013 in the above example.
The byte address is used to specify the relay area in terms of bytes for source and destination by the application instructions.

B. Absolute Address

A consecutive number is assigned to each group of I/O relays, link relays, internal relays, and latched relays. There are unused relay numbers in each group. For example the relay numbers between the I/O relays and the link relays (1000 to 1777). The data memory RAM has relay numbers that are sequential and without gaps.

The absolute address is the byte term address represented without breaks between the groups.

].000~].077
].100~].177*
].200~].277
].300~].377*
].400~].477
].500~].577*
].600~].677*
].700~].777

I/O relay	].000 ~].077	000 ~ 077
Link relay	].200 ~].277	100 ~ 177
Internal relay	].400 ~].477	200 ~ 277
Latched relay	].700 ~].777	300 ~ 377

* = Relay numbers that are not user accessible.

1. The absolute address is used for securing the latched relay area and output retaining area when an error occurs.
2. An absolute address must be used in some of the application instructions.

Basic Relay Instructions

1. Store (STR)/Out (OUT) Instructions

If each logic line starts with a N.O. contact, use the STR instruction.

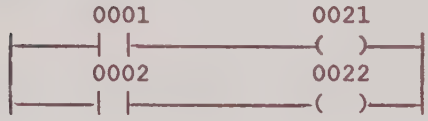
Ex.1

Address	Instruction	Data
0000	STR	0000
0001	OUT	0020

Keystrokes:

1. *, *, PROG MODE, */SET	Puts the 0185 in the Prog mode
2. CLEAR	Brings up address 0000
3. STR, 0, 0, 0, 0, ENTER	Programs N.O. contact at #0000
4. NEXT ADRS	Steps to address 0001
5. OUT, 0, 0, 2, 0, ENTER	Programs coil # 0020

Ex.2

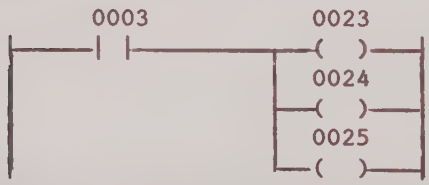


Address	Instruction	Data
0002	STR	0001
0003	OUT	0021
0004	STR	0002
0005	OUT	0022

Keystrokes:

1. NEXT ADRS	Steps to address #0002
2. STR, 0, 0, 0, 1, ENTER, NEXT ADR	Programs N.O. contact #0001
3. OUT, 0, 0, 2, 1, ENTER, NEXT ADR	Programs coil #0021
4. STR, 0, 0, 0, 2, ENTER, NEXT ADR	Programs N.O. contact #0002
5. OUT, 0, 0, 2, 2, ENTER, NEXT ADR	Programs coil #0022

Ex.3



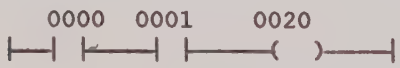
Address	Instruction	Data
0006	STR	0003
0007	OUT	0023
0008	OUT	0024
0009	OUT	0025

Example 3 works because the OUT instruction does not effect the previous instructions.

2. AND (AND) Instruction

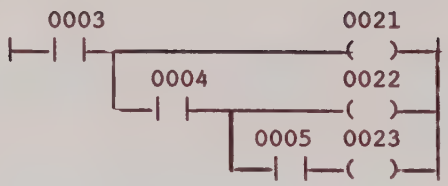
N.O. contacts in series are processed by the AND instruction.

Ex.1



Address	Instruction	Data
0000	STR	0000
0001	AND	0001
0002	OUT	0020

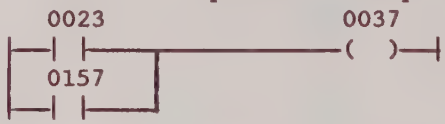
Ex.2



Address	Instruction	Data
0000	STR	0003
0001	OUT	0021
0002	AND	0004
0003	OUT	0022
0004	AND	0005
0005	OUT	0023

3. OR (OR) Instruction

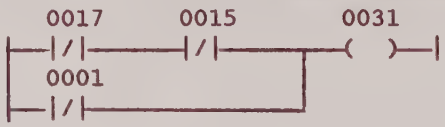
N.O. contacts in parallel are processed by the OR instruction.



Address	Instruction	Data
0150	STR	0023
0151	OR	0157
0152	OUT	0037

4. STORE NOT (STR NOT)/ OR NOT (OR NOT)/ AND NOT (AND NOT)

If a line starts with a N.C. contact, or a parallel/series circuit using a N.C. contact use the NOT instructions.

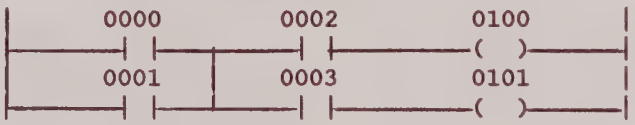


Address	Instruction	Data
0000	STR NOT	0017
0001	AND NOT	0015
0002	OR NOT	0001

5. Ladder Logic Diagram Cautions

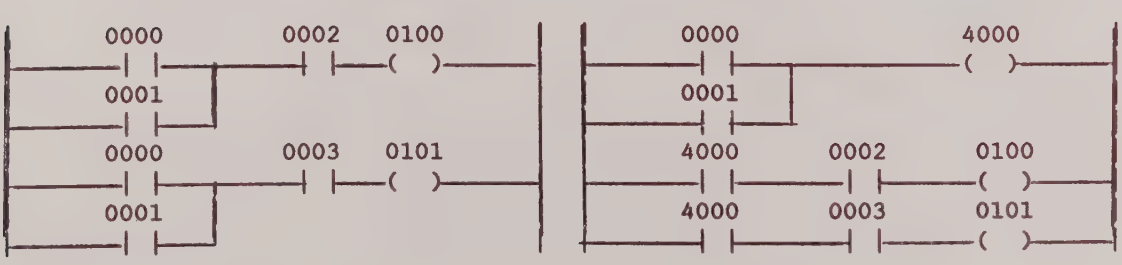
The PC 0185 solves logic sequentially. Therfore, some ladder logic diagrams and statement logic diagrams are not equivalent.

Example:



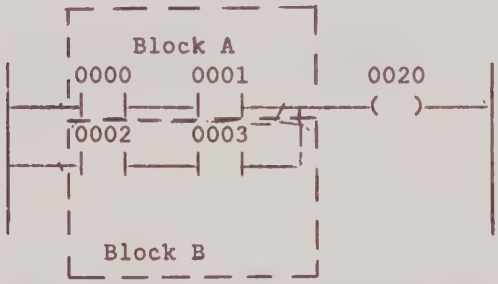
This must be redrawn for the PC 0185:

A OR B



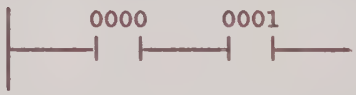
6. Programming Series/Parallel Circuits

During the processing of serial/parrallel circuits or application instructions that have multiple numbers of inputs, the intermediate results are stored in a stack register. Prior to programming the serial/parralel circuit, it should be divided into block. By using the STR function, you create these minor blocks and shift them into the accumulator and stack register. Executing the (AND STR)/(OR STR) functions pulls the minor blocks out of the stack, linking the minor blocks into a completed ladder program.



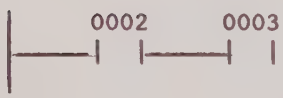
Address	Instr	Data	
0000	STR	0000	
0001	AND	0001	Block A
0002	STR	0002	
0003	AND	0003	Block B
0004	OR STR		
0005	OUT	0020	

NOTE: The contents of block A are programmed in the accumulator.

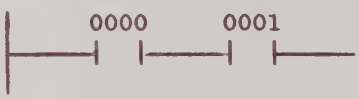


ACC

Next the contents of block B are stored in the accumulator, pushing block A into the stack.

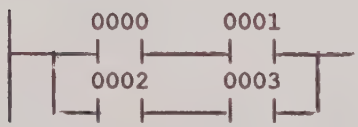


ACC



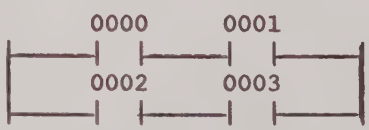
STACK

The (OR STR) function pulls block A out of the stack and block B is ORed with it.



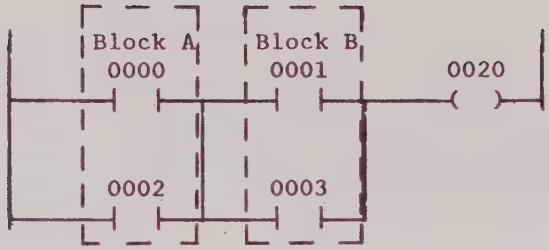
ACC

ie;



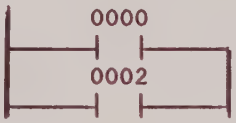
ACC

EX. 2

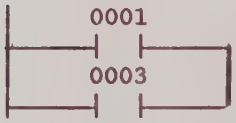


Address	Instruction	I/O
0000	STR	0000
0001	OR	0002 Block A
0002	STR	0001
0003	OR	0003 Block B
0004	AND STR	
0005	OUT	0020

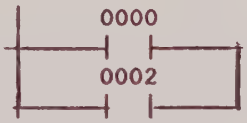
NOTE: The contents of Block A are stored in the accumulator.



Next the contents of Block B are stored in the accumulator and Block A is pushed into the stack.



ACC

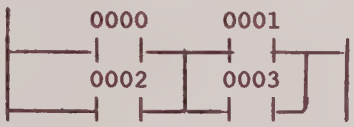


STACK

The (AND STR) pulls Block A out of the stack and attaches it in front of Block B, connecting the top rung.

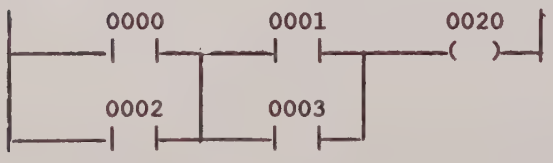
ACC

ie;



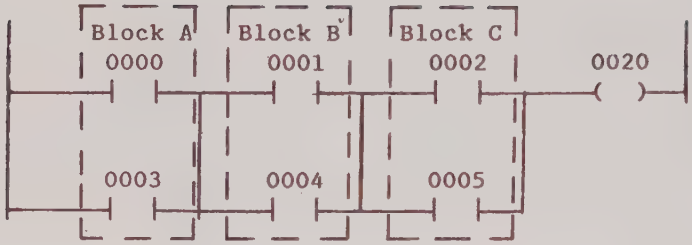
ACC

The (out) function programs the coil.



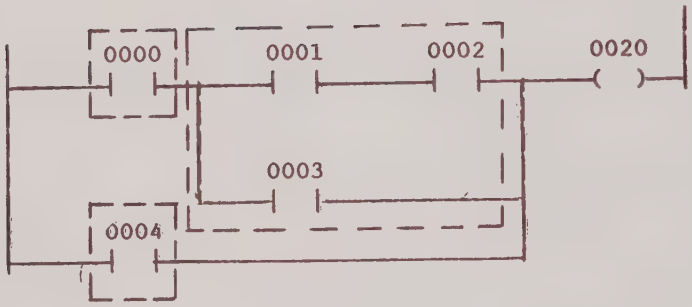
ACC

EX. 3



Address	Instruction	I/O
0000	STR	0000
0001	OR	0003
0002	STR	0001
0003	OR	0004
0004	STR	0002
0005	OR	0005
0006	AND STR	
0007	AND STR	
0008	OUT	0020

EX 4



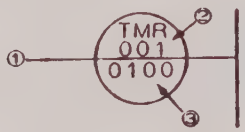
Address	Instruction	I/O
0000	STR	0000
0001	STR	0001
0002	AND	0002
0003	OR	0003
0004	AND STR	
0005	STR	0004
0006	OR STR	
0007	OUT	0020

TMR (timer instruction)

The instruction decrements the timer at 0.1 second intervals. Timing does not take place while the start input if OFF. When the timer's start input is ON, the current timer value is decremented by 1 every 0.1 seconds. When the current value reaches 0, the TMR contact turns ON; and remains ON as long as the start input is ON.

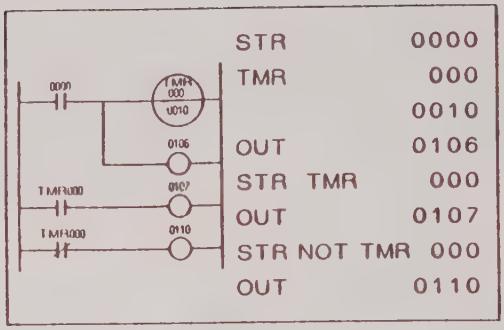
Start input	Current value	TMR contact
OFF	Preset value	OFF
ON (current value > 0)	Decremented by one every 0.1 second	OFF
ON (current value = 0)	0	ON
ON → OFF (current value > 0)	Reset to the preset value	OFF
ON → OFF (current value = 0)	Reset to the preset value	ON → OFF

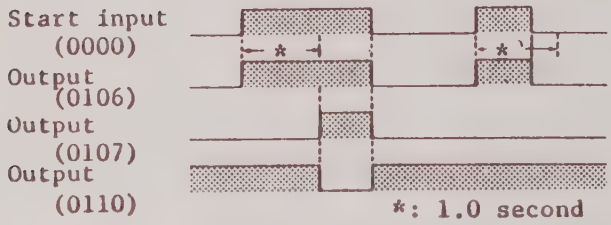
Symbol representing the TMR instruction



Symbol representing the TMR contact

①	Start input	(start with ON state of signal)
②	TMR number	000 ∿ 177 ... Used common for CNT and MD.
③	Preset value	0001 ∿ 1999 (BCD) ... In an increment of 0.1 second (000.1 ∿ 199.9 seconds)





Because the TMR, CNT, and MD instructions use the same memory area, a number can only be used once in a program.

The TMR contact can be the same number as the TMR number; and any number of a-type (N.O.) contacts or b-type contacts (N.C.) can be used.

The current value of the timer is held in 256 bytes of b.000 to b.377.

The timer is reset on power up. The preset value will be used for the current value when the power is turned on (if the start input is ON).

By setting the timer's reset condition in the system memory (#201), it is possible to store the state at the time of power interrupt.

It is possible to implement the power failure timer and timers differing in their start and reset conditions by making use of the contact 7360 (0.1 second clock) and the CNT instruction.

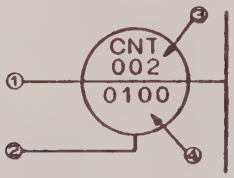
CNT (counter instruction)

The CNT instruction decrements the counter by one on every low to high transition of the counter input.

No counting takes place if the reset input is ON. As the counter input changes from OFF to ON while the reset input is OFF, the current CNT value decrements by one. When the current value is 0, the CNT contact is turned ON; and remains ON as long as the reset input is OFF.

Reset input	Current value	CNT contact
ON	Preset value	OFF
OFF (current value > 0)	Decrements by one every time the counter input changes from OFF to ON.	OFF
OFF (current value = 0)	0	ON
OFF → ON (current value > 0)	Reset to the preset value	OFF
OFF → ON (current value = 0)	Reset to the preset value	ON → OFF

Symbol representing the CNT instruction

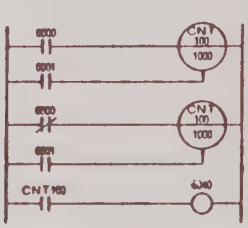


Symbol representing the CNT contact

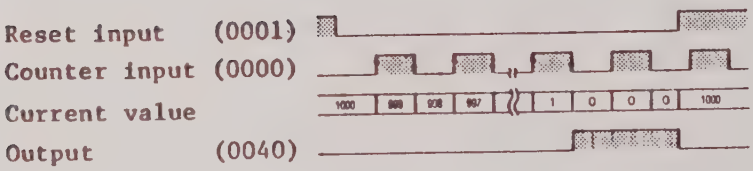
①	Counter input	(senses OFF to ON transition)
②	Reset input	(reset with ON)
③	CNT number	000 ∿ 177 (octal number) ... Used commonly with TMR and MD.
④	Preset value	0001 ∿ 1999 (BCD)

Because the TMR, CNT, and MD instructions use the same memory area, a number can only be used once in a program.

Rising and falling edge triggered counter
(The counter in this example counts each time the counter input changes from OFF to ON or ON to OFF).



Instruction	
STR	0000
STR	0001
CNT	100
	1000
STR NOT	0000
STR	0001
CNT	100
	1000
STR CNT	100
OUT	0040



The CNT contact can be the same number as the CNT number; and any number of a-type (N.O.) contacts or b-type contacts (N.C.) can be used.

The current value of the counter is held in 256 bytes of b.000 to b.377

When the counter reaches zero the reset input must be turned on to start counting again.

When both the counter input and reset input go ON simultaneously, the reset input takes precedence.

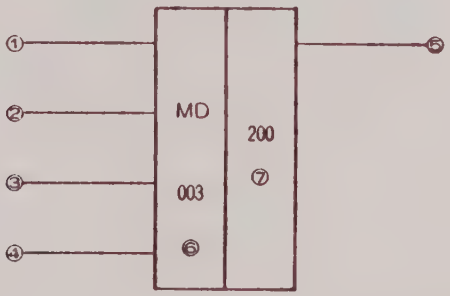
If the power is interrupted, the counter retains its current value. The current value will be reset if the reset input is turned ON at power on. Setting the reset condition in the system memory (#202) will give control over the state of the reset signal at power up.

MD (maintenance display)

The MD instruction is used to:

(a) monitor the operation condition of the controlled device,
(b) trace the cause of trouble using the programmer,
(c) send the output signal to an external source.

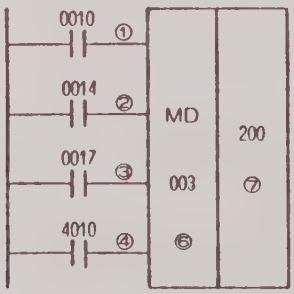
A. Description of the symbol:



① ② ③	Input information	External output contact information used in conjunction with the MD data of ⑦, and relay area 0000 ∿ 7377 and TMR and CNT contact may be used.
④	Output direct condition	The input is used to direct whether the contact information of ①, ②, and ③ and MD data of ⑦ should be outputted to the data memory or relay area of the MD number specified in ⑥, and relay area 0000 ∿ 7377 and TMR and CNT contact may be used. Output becomes active when ON. No change occurs in the contact information and MD information even if it turned OFF.
⑤	MD expansion output	There is no need program the MD instruction with the condition in ④ above, when using the MD instruction continuously with the same output direct condition.

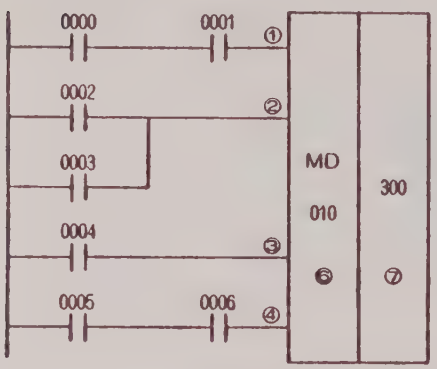
⑥	MD number	The MD instruction uses the TMR and CNT current value storage area (b.000 ∿ b.377) or relay area (0000 ∿ 7377) for the storage of contact information of ① , ② , and ③ and MD data of ⑦ . (1) When the TMR or CNT area is used It should be programmed using the number of 000 ∿ 177 and the information must be monitored on the programmer, etc. NOTE: It is not possible to use the same number as used for the TMR and CNT. (2) When the relay area is used It should be programmed in terms of bytes address ⊐.×××. Specifying ⊐.000 by programming will reserve two bytes of ⊐.000 and .001 for the MD area. Use of the area connected with the output unit will permit external output (display) of the contact information of ① , ② , and ③ and the MD data of ⑦ .
⑦	MD data	Any number of 000 ∿ 999 may be used in the BCD mode. It should be programmed in connection with the process number, relay number, external device number, etc.

B. MD Instruction Programming Procedure




```
STR     0010  ---①
STR     0014  ---②   Input information
STR     0017  ---③
STR     4010  ---④   Output direction
MD       003  ---⑤   MD number
         200  ---⑦   MD data
```

Complicated conditional statements may be handled for both the input information and the output directing condition.

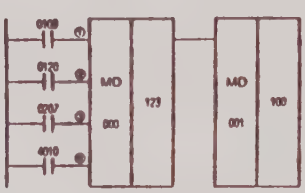



```
STR   0000 ) -①
AND   0001 )
STR   0002 ) -②   Input
OR    0003 )      information
STR   0004 —③
STR   0005 ) -④Output direction
AND   0006 )
MD     010 —⑥MD number
       300 —⑦MD data
```

Accumulator and stack register transactions during the MD instruction.

	Accumulator	Stack register		
	ACC	S1	S2	S3
STR 0000	0000			
AND 0001	0000 0001			
STR 0002	0002	0000 0001		
OR 0003	0002 0003	0000 0001		
STR 0004	0004	0002 0003	0000 0001	
STR 0005	0005	0004	0002 0003	0000 0001
AND 0006	0005 0006	0004	0002 0003	0000 0001
MD 010 300	Output direction ④	Input information ③	Input information ②	Input information ①

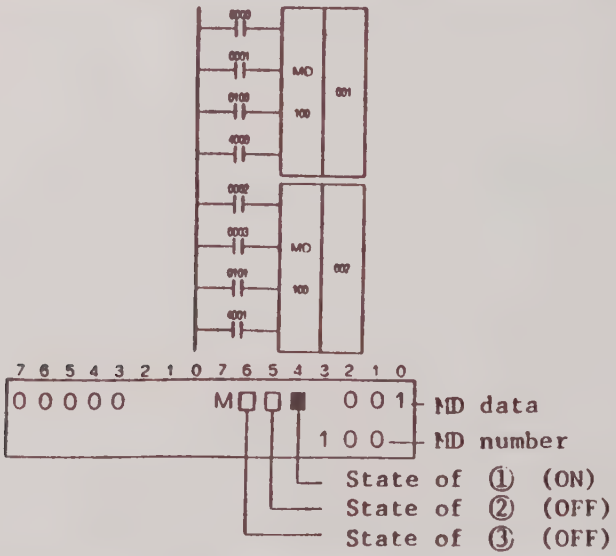
The state of the accumulator and stack register are not changed after the execution of the MD instruction. This permits stacking of MD instructions.




```
STR   0100
STR   0120
STR   0207
STR   4010
MD     000
       123
MD     001
       100
```

C. Monitoring MD Information

The information can be monitored on the programmer in the following manner:



The next information can be obtained from the programmer display.

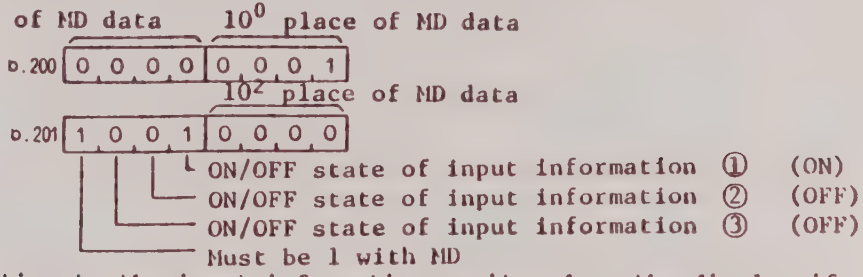
Because the MD data is 001:

a. 4001 is OFF with ON state of internal contact 4000

b. Input information displayed:
   1.....0000 (ON)
   2.....0001 (OFF)
   3.....0100 (OFF)

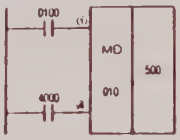
EXAMPLE

The MD information in MD 100 is stored in b.200 and b.201 of the data memory.


10^1 place of MD data
10^0 place of MD data
b.200 0 0 0 0 0 0 0 1
10^2 place of MD data
b.201 1 0 0 1 0 0 0 0
ON/OFF state of input information ① (ON)
ON/OFF state of input information ② (OFF)
ON/OFF state of input information ③ (OFF)
Must be 1 with MD

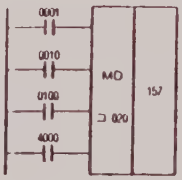
Pay attention to the input information monitored on the display if 1, 2, or 3 is not used.

In the following case input information 1 is stored in stack register 21. The output direction condition (4) is stored in the accumulator when MD 010 is executed. Since the intermediate result of the preceding operation remains in the stack registers (S2 and S3) they are not significant as MD information.



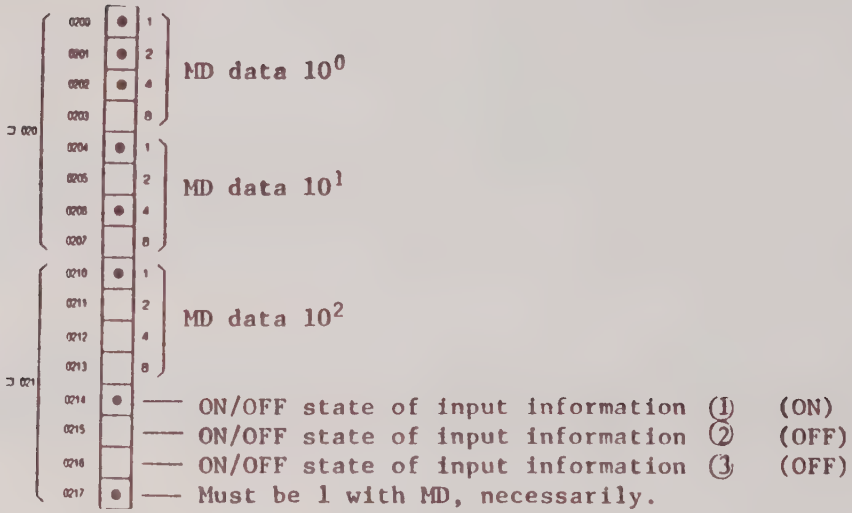
D. MD Information as an External Output

When the data memory relay area is specified in terms of byte address, instead of MD number, it is possible to send the MD output to an external source or transmit it to another PC using a data link function.



When].020 is specified the MD information is stored in two bytes (].020 and].021). The output unit is connected to].020 and].021.

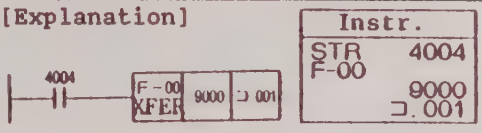
The • mark represents the ON state.



F-00 XFER	Transfer one byte of data

Symbol	
Function	The contents of register S (one byte data) are transferred to register D.
Operation	S→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Contents of the register S
Flag	Unchanged

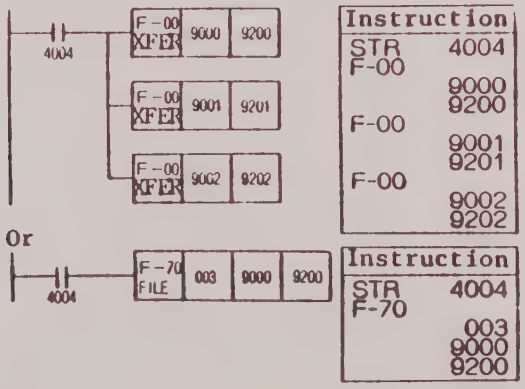
[Explanation]



When the input condition 4004 changes from OFF to ON, the contents of register 9000 are transferred to].001.

NOTE-1:].734 ~].737 is the special area.

REFERENCE: To transfer more than two bytes in succession, either repeat the F-00 instruction as shown below or use the n-bytes batch transfer instruction F-70.



F-01 BCD	Transfer BCD constant

Symbol	
Function	A 2-digit BCD constant "n" is transferred to register D.
Operation	N→D
Range of "n" applicable	00~99
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	n(00~99)
Flag	Unchanged

[Explanation]

Instr.	
STR	4004
F-01	
	15
	9100

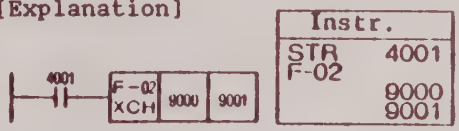
When the input condition 4004 changes from OFF to ON, the BCD constant "15" is transferred to register 9100.

NOTE-1:].734 ~].737 is the special area.

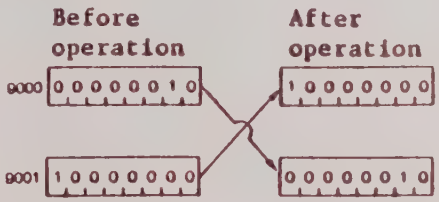
F-02 XCH	Exchange Data between registers (eXCHange)

Symbol	
Function	The contents of register D_1 are exchanged with the contents of register D_2
Operation	$D_1 \leftrightarrow D_2$
Range of D_1 applicable	00~99
Range of D_2 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D_1	Contents of the register D_2
Contents of D_2	Contents of the register D_1
Flag	Unchanged

[Explanation]



When the input condition 4004 changes from OFF to ON, the contents of register 9000 are exchanged with the contents of register 9001.



NOTE-1:].734 ~].737 is the special area.

F-03 →BIN	Convert BCD into binary

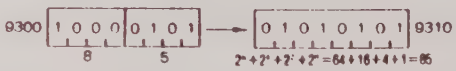
Symbol	
Function	The contents of the register S (8-bit data assumed to be BCD), are register D.
Operation	S→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	° Operational result ° Remains unchanged if the contents of the register S is not a BCD code
Flag	Con. of reg.S | a | b | c | d BCD | | 0 | | | If not BCD | 0 | 1 | 0 | 0 a:Non-carry 7354 b:Error 7355 c:Carry 7356 d:Zero 7357

[Explanation]

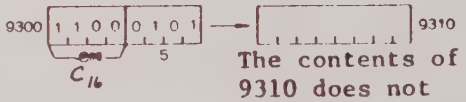
Instr.	
STR	4006
F-03	9300
	9310

° When the input condition 4006 changes from OFF to ON, the contents of register 9300 (8-bit data, assumed to be BCD), are converted into a binary equivalent. Its result is transferred to register 9310. The contents of register 9300 remains unchanged. If the contents of 9300 is other than BCD code, the contents of register 9310 stay the same as before and the error flag (7355) IS SET TO "1".

° Transition of register contents and flags



Non-carry 7354	Error 7355	Carry 7356	Zero 7357
0	0	0	0



The contents of 9310 does not change.

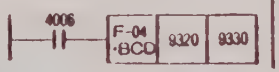
Non-carry 7354	Error 7355	Carry 7356	Zero 7357
0	1	0	0

NOTE-1:].734 ~].737 is the special area.

F-04 →BCD	Convert binary into BCD

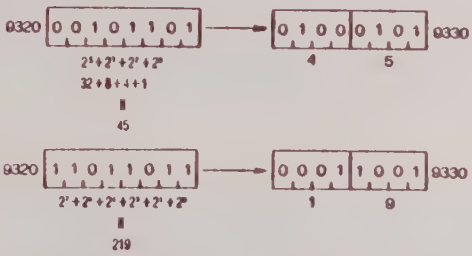
Symbol	
Function	The contents of register S (8-bit data assumed to be binary), are converted into a BCD equivalent, then the result is stored in register D.
Operation	N→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Operational result
Flag	Unchanged

[Explanation]



Instr.	
STR	4006
F-04	
	9320
	9330

° When the input condition 4006 changes from OFF to ON, the contents of the register 9320 (8-bit data assumed to be binary), are converted into a BCD equivalent. Its result is transferred to register 9330. The contents of register 9320 remains unchanged. If the BCD number converted should exceed "100", the value above "100" will be disregarded.



NOTE-1:].734 ~].737 is the special area.

F-05	Demultiplex one byte of data
DMPX	(DeMultiPleXer)

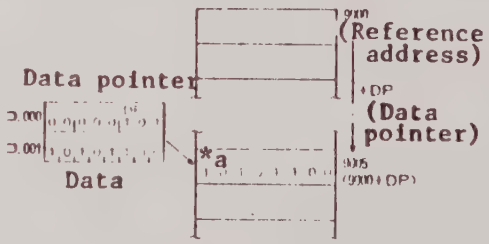
Symbol	F-05 DMPX | S | D
Function	The contents of register S+1 are transferred to register D (reference address) plus the contents of register S (data pointer).
Operation	S + 1 → <S> + D ↓ Data pointer (DP) Reference address
Range of S applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000(NOTE-1) b.000 9000 30000 30400 31000 31400 32000 32400 33000 33400
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S,S+1	Unchanged
Contents of D	Unchanged
Contents of D+<s>	Contents of the register S + 1
Flag	Unchanged

[Explanation]

Instr.	
STR	4005
F-05	].000 9000

° The following transfer takes place when the input condition 4005 changes from OFF to ON. The data in].000+1 (that is the data in].001) are transferred to the address 9005 which is the reference address (9000) plus the contents of the data pointer (005($_8$) in].000).

The data pointer is represented by an octal number (000 to 377) In the above example, it would be possible to demultiplex data to address, 9000 ~ 9377, by varying the data pointer value in reference to the reference address (9000).



*a: Designation register

NOTE-1:].734 ~].737 is the special area.

(1) If S is assumed to be].733, S+1 will be in the specific area.

(2) Be careful that the contents of S (data pointer) do not exceed 333( ) in using].000 for D, because the absolute address of 334($_8$) is].734.

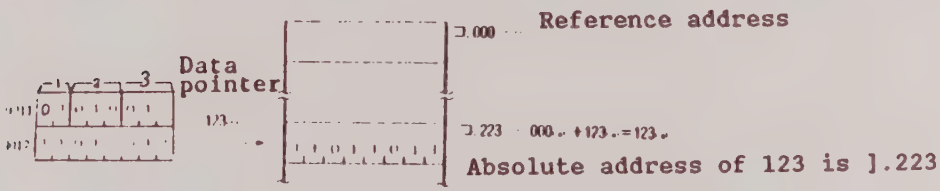
NOTE-2: File register, 30000 ~ 33777, can be used when the file register capacity has been set to the system memory (#205).

NOTE-3: Although it is possible to program other than the reference address into D, the programmable controller references the starting address of the block that contains the specified address for D.

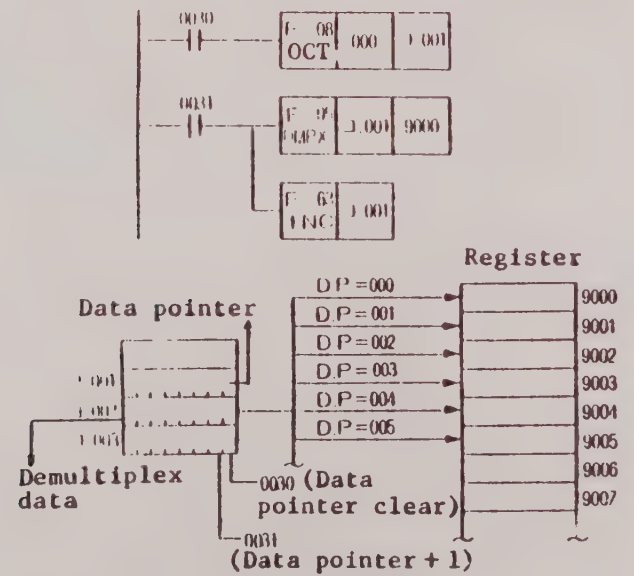
(Ex)

D representation	Reference address operation-wise
].400	].000
b.210	b.000
9105	9000
33210	33000

NOTE-4: When].xxx area is used for D, the address of the demultiplexed register is determined by the absolute address that has].000 for the reference address.



REFERENCE: A programming example to change the demultiplexing address by varying the data pointer.



1 When 0030 is changed from OFF to ON, the octal constant 000 is transferred to].001. (Data pointer 000)

2 When 0031 is turned from OFF to ON, the contents of].002 are transferred to (9000 + 000 = 9000). The contents of].001 change to 0001 as it is incremented plus one by the F-63 instruction.

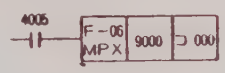
3 When 0031 is changed from OFF to ON again, the contents of].002 are transferred to (9000 + 001 = 9001). The contents of].001 change to 002 as it is incremented plus one by the F-63 instruction.

Thereafter, the contents of].002 are demultiplexed to registers up to 9377.

F-06 MPX	Multiplex one byte of data (MultiPleXer)

Symbol	
Function	The contents of register whose address is designated by the register S (reference address) plus the contents of register D (data pointer) are transferred to the register D + 1.
Operation	S + <D> → D + 1 | | | |--→Data pointer (DP) |--→Reference address
Range of S applicable	].000 b.000 9000 30000 30400 31000 31400 32000 32400 33000 33400
Range of D applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Unchanged
Contents of D+1	Contents of the register S + <D>
Flag	Unchanged

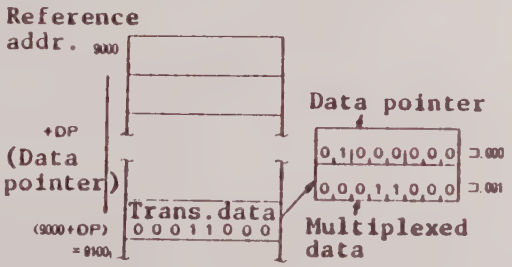
[Explanation]



Instr.	
STR	4005
F-06	9000 D.000

The following transfer takes place when the input condition 4005 changes from OFF to ON. The contents of address 9100 which is the reference address 9000 plus the contents of the varying the data pointer (100 ($_8$) in].000) are transferred to].000+1 (].001).

The data pointer is represented by octal number (000 to 377). In the above example, it would be possible to multiplex data to address, 9000 ~ 9377, by varying the data pointer value in reference to the reference address.



NOTE-1:].734 ~].737 is the special area.

(1) If D is assumed to be].733, D+1 be in the special area.

(2) The contents of D (data pointer) can not exceed 333($_8$) in using].000 for S, because the absolute address of 334 ($_8$) is].734.

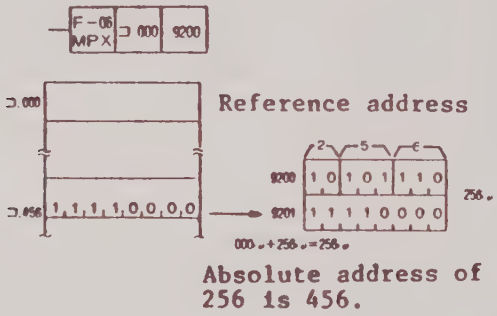
NOTE-2: File register, 30000 ~ 33777, can be used when the file register capacity has been set to the system memory (#205).

NOTE-3: Although it is possible to program other than the reference address into S, the programmable controller reference's the starting address of the block that contains the specified address for D.

(Ex)

D representation	Reference address operation-wise
].211	].000
b.106	b.000
9023	9000
31257	31000

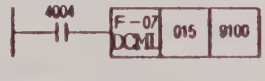
NOTE-4: When].xxx area is used for S, the address of the multiplexed register is determined by the absolute address that has].000 for the reference address.



F-07 DCML	Transfer decimal constant

Symbol	F-07 DCML | n | D
Function	The decimal constant "n" is transferred to the register D.
Operation	n→D
Range of "n" applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	n(000~255)
Flag	Unchanged

[Explanation]



Instr.	
STR	4004
F-07	
	015
	9100

° When the input condition 4004 changes from OFF to ON, the decimal constant 15 is transferred to register 9100.

° Register 9100 has the following binary code representation

NOTE-1:].734 ~].737 is the special area.

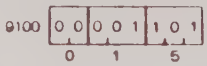
F-08 OCT	Transfer octal constant

Symbol	F-08 OCT | n | D
Function	The octal constant "n" is transferred to register D.
Operation	n→D
Range of "n" applicable	000 ~ 377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	n(000~377)
Flag	Unchanged

[Explanation]

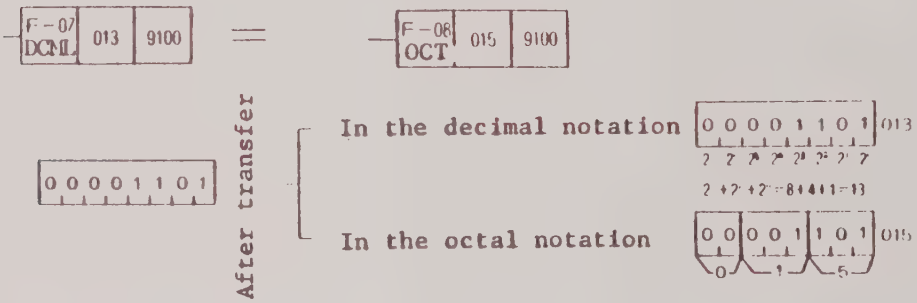
Instr.	
STR	4004
F-08	
	015
	9100

- When the input condition 4004 changes from OFF to ON, the octal constant 015 is transferred to register 9100.
- Register 9100 has the following binary code representation



NOTE-1:].734 ~].737 is the special area.

REFERENCE: Though the F-07 (transfer decimal constant) handles a decimal number and the F-08 (transfer octal constant) an octal number, the contents of the register after the transfer are represented in binary code for both instructions.


F-07 DCML 013 9100
=
F-08 OCT 015 9100
0 0 0 0 1 1 0 1
After transfer
In the decimal notation
0 0 0 0 1 1 0 1 013
2 + 2 + 2 = 8 + 4 + 1 = 13
In the octal notation
0 0 0 0 1 1 0 1 015
0
1
5

When the F-08 instruction is used to preset the data pointer for the F-05 (demultiplex) and F-06 (multiplex) instructions, the data memory address will be in octal.

F-09 INV — Invert 8-bit data

Symbol	
Function	The contents of register S are inverted and stored in register D.
Operation	S→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	Inversion of register S contents
Flag	Unchanged

[Explanation]

° When the input condition 4002 changes from OFF to ON, the 8-bit contents of register 9000 are inverted and its result is stored in register 9003.

The contents of register 9000 remain unchanged.

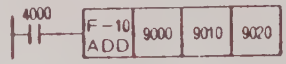
NOTE-1:].734 ~].737 is the special area.

F-10 ADD	Add registers (BCD 2-digit) (ADD)

Symbol	F-10 ADD | S_1 | S_2 | D
Function	The contents of register S_1 are added to the contents of register S_2 (BCD 2-digit). Its result is stored in register D.
Operation	$S_1+S_2 \rightarrow D$
Range of S_1 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1	Unchanged
Contents of S_2	Unchanged
Contents of D	° Low order two digits of the operational result ° Unchanged when the contents of registers S_1 or S_2 are not BCD code.
Flag	a:Non-carry 7354 b:Error 7355 c:Carry 7356 d:Zero 7357

Result	a	b	c	d
0	1	0	0	1
1~99	1	0	0	0
100	0	0	1	1
101 and above	0	0	1	0
When S_1 and S_2 are not BCD code	0	1	0	0

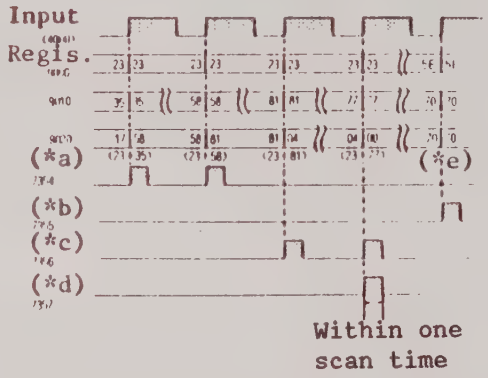
[Explanation]



Instr.	
STR	4000
F-10	
	9000
	9010
	9020

° When the input condition 4000 changes from OFF to ON, the contents of register 9000 are added to the contents of register 9010. Its result is stored in register 9020. The contents of registers 9000 and 9010 remain unchanged.

° Transition of operational result and flags.



Remains valid until the flag-affecting instruction appears in the program.

(*a):Non-carry flag
(*b):Error flag
(*c):Carry flag
(*d):Zero flag
(*e):This does not operate because the contents of 9000 are not in BCD.

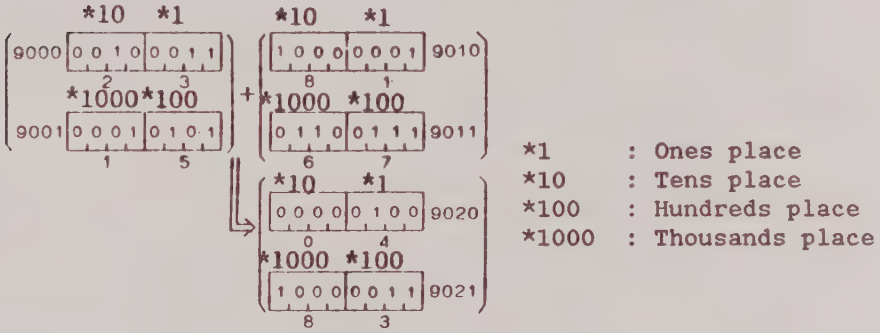
NOTE-1:].734 ~].737 is the special area.

NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

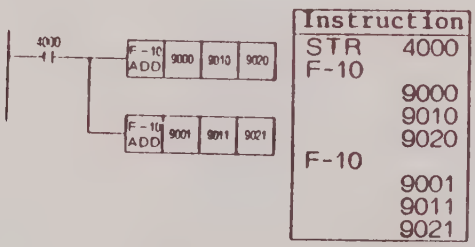
NOTE-3: If the contents of S_1 or S_2 were numbers other that BCD, error flag (7355) is set active (1) and no addition will be done.

(Ex) 1110 is not BCD

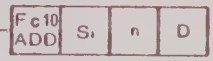
REFERENCE: In case more than 3 digits of BCD to be added successively, the F-10 instruction can be used over and over. When the F-10 instruction is programmed repeatedly, the contents of the carry flag (7356) are also added after the second instruction. For the F-10 instruction that appears after the STR instruction, the contents of the carry flag (7356) are not added.



- The above example shows 1523 + 6781 = 8304
- If programmed from the low order digit, the carry information will be carried on to a higher digit.



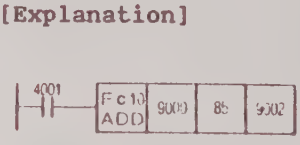
Fc-10 ADD	Add registers with 2-digit BCD constant (ADD)

Symbol	
Function	The contents of register S_1 are added with a 2-digit BCD constant "n" and is stored in register D.
Operation	S_1+n→D
Range of S_1 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of "n" applicable	00~99
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1	Unchanged
Contents of S_2	Unchanged
Contents of D	° Low order two digits of the operational result ° Unchanged when the contents of registers S_1or S_2 are not BCD

Flag	Result	a	b	c	d
	0	1	0	0	1
	1~99	1	0	0	0
	100	0	0	1	1
	101 and above	0	0	1	0
	When S_1 is not BCD code	0	1	0	0

a:Non-carry 7354
b:Error 7355
c:Carry 7356
d:Zero 7357

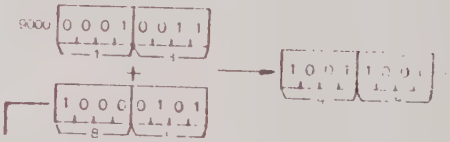
[Explanation]



Instr.	
STR	4001
Fc10	
	9000
	85
	9002

When the input condition 4001 changes from OFF to ON, the contents of register 9000 are added to the BCD constant 85. Its result is stored in register 9002.

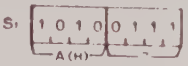
It operates with the same timing as the F-10 instruction.



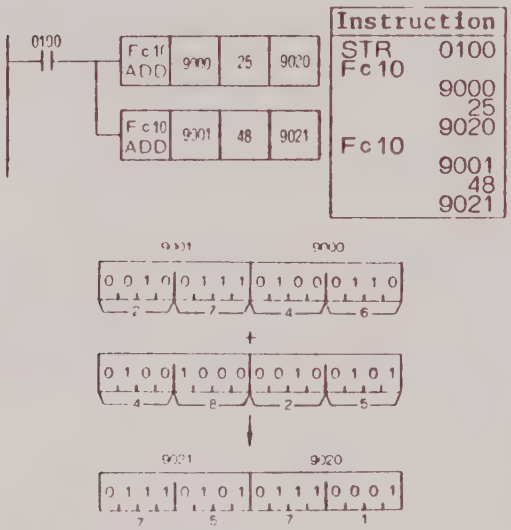
NOTE-1:].734 ~].737 is the special area.

NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

NOTE-3: If the contents of S_1 or S_2 were numbers other that BCD, the error flag (7355) is set active (1) and no addition will be done.

(Ex) 1010 is not a BCD number 

REFERENCE: As with the F-10 instruction, it is possible to add more than three digits of BCD value.



F-11 SUB	Subtract register from register in BCD (2 digits) (SUBtract)

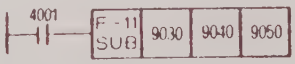
Symbol	
Function	The contents of register S_1 are subtracted from register S_2 in the BCD 2 digits. Its result is stored in the register D.
Operation	$S_1-S_2 \rightarrow D$
Range of S_1 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of S_2 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1	Unchanged
Contents of S_2	Unchanged
Contents of D	° Operational result ° Not changed when the contents of registers S_1 or S_2 are not BCD code.

Flag	Result	a	b	c	d
	0	1	0	0	1
	1~99	1	0	0	0
	Negative value	0	0	1	0
	When S_1 or S_2 are not BCD	0	1	0	0

a:Non-carry 7354
b:Error 7355
c:Carry 7356
d:Zero 7357

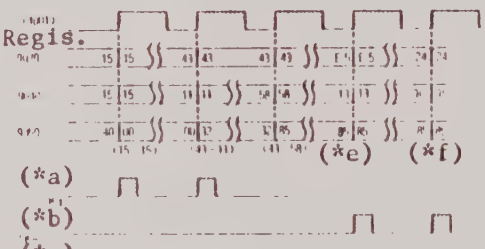
[Explanation]

Instr.	
STR	4001
F-11	9030
	9040
	9050



When the input condition 4001 changes from OFF to ON, the contents of register 9030 is subtracted from the contents of register 9040. Its result is stored in register 9050. The contents of registers 9030 and 9040 remain unchanged.

Transition of operational result and flags



Within one scan time
Remains valid until a flag-affecting instruction is met in the program.

(*a):Non-carry flag
(*b):Error flag
(*c):Carry flag
(*d):Zero flag
(*e):No subtraction is done because 9030 is not in a BCD number.
(*f):No subtraction is done because 9040 is not a BCD number

NOTE-1:].734 ~].737 is the special area.

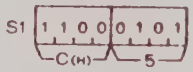
NOTE-2: If the contents of S_1 is smaller than the contents of S_2, the result is the complement of 100.

(Ex) 2 - 3 = 99

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

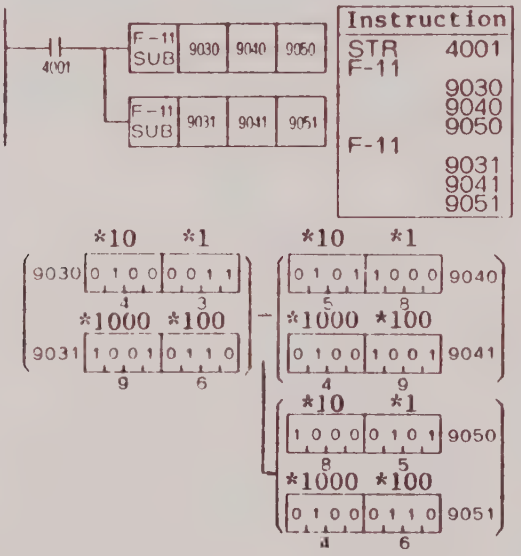
NOTE-4: If the contents of S_1 or S_2 are numbers other than BCD, the error flag (7355) is activated, and no subtraction is done. (The contents of D remain unchanged.)

(Ex)



1100 is prohibited in the BCD system.

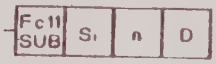
REFERENCE: In case more than 3 digits of BCD have to be subtracted successively, the F-11 instruction can be used over and over. When the F-11 instruction is programmed repeatedly, the contents of the carry flag (7356) are also subtracted after the second instruction. For the F-11 instruction that appears after the STR instruction, the contents of the carry flag (7356) are not subtracted.



*1 : Ones place
*10 : Tens place
*100 : Hundreds place
*1000: Thousands place

- The above example shows 9643 - 4958 + 4685.
- If programmed from the low order digit, the carry information will be carried on to a higher digit.

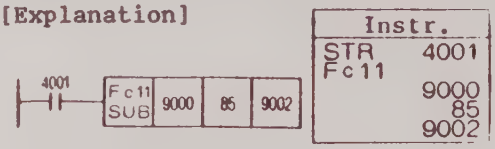
Fc11 SUB	Subtract register from 2-digit constant (SUBtract)

Symbol	
Function	The contents of register S_1 are subtracted from 2 digit BCD constant "n". Its result is stored in the register D.
Operation	S_1-n→D
Range of S_1 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of n applicable	00~99
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1	Unchanged

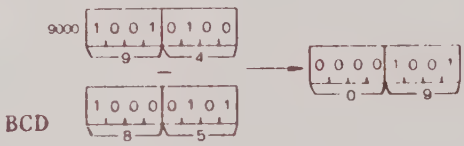
Flag	Result	a	b	c	d
	0	1	0	0	1
	1~99	1	0	0	0
	Negative value	0	0	1	0
	When S_1 or S_2 are not BCD	0	1	0	0

a:Non-carry 7354
b:Error 7355
c:Carry 7356
d:Zero 7357

[Explanation]



When the input condition 4001 changes from OFF to ON, the contents of register 9000 are subtracted from the BCD constant 85. Its result is stored in register 9002. It operates with the same timing as the F-11 instruction.



BCD constant 85

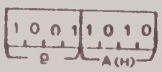
NOTE-1:].734~].737 is a special area.

NOTE-2: If S_1 is smaller than "n", the result will be the complement of 100.

(Ex) 2 - 3 = 99

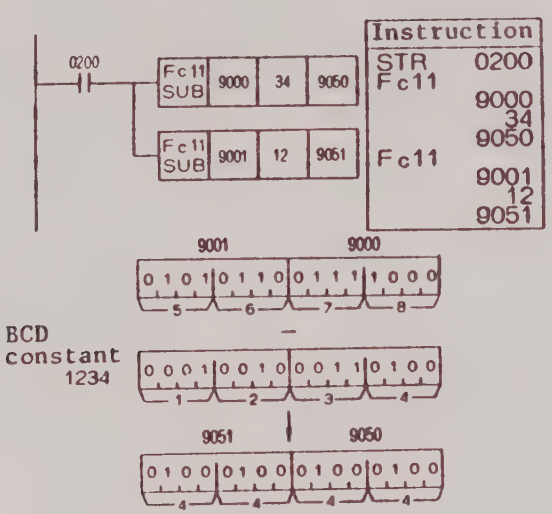
NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

NOTE-4: If the contents of S was a number other than BCD, the error flag (7355) is set active (1) and no subtraction will be done.

(Ex) 

1010 is not a BCD number.

REFERENCE: As with the F-11 instruction, it is possible to subtract more than three digits of BCD value.



F-11	Subtract register from register in BCD (2 digits)
SUB	(SUBtract)

Symbol	F-12 CMP S1 S
Function	The contents of register S_1 are compared with the contents of register S_2.
Operation	S_1<=>S_2
Range of S_1 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of S_2 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	When the input signal is in the ON state (not limited to the OFF to ON transition.
Contents of S_1	Unchanged
Contents of S_2	Unchanged

Flag	Result	a	b	c	d
	S_1>S_2	1	0	0	0
	S_1=S_2	1	0	0	1
	S_1<S_2	0	0	1	0

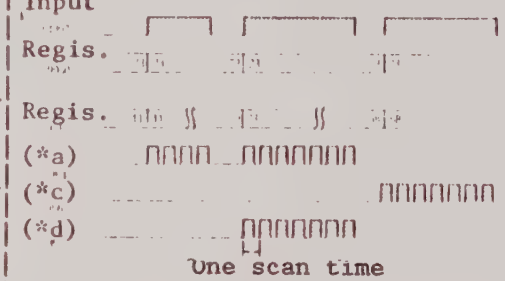
a:Non-carry 7354
b:Error 7355
c:Carry 7356
d:Zero 7357

[Explanation]

Instr.	
STR	4003
F-12	9000
	9010

When the input condition 4003 is ON, the contents of register 9000 are compared with the contents of register 9010. Its result is are set in the non-carry flag (7354), carry flag (7356), and zero flag (7357). Registers 9000 and 9010 remain unchanged after this operation.

° Transition of register contents



As long as the input condition is ON, comparison is done every scan, and the zero flag is turned ON.

(*a):Non-carry flag
(*c):Carry flag
(*d):Zero flag

NOTE-1:].734 ~].737 is the special area.

NOTE-2: As long as the input signal is ON, comparison is done every scan cycle.

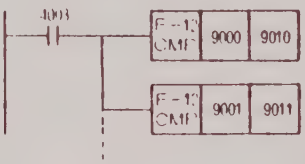
NOTE-3: The error flag (7355) is 0 at all times.

NOTE-4: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

REFERENCE: To make a comparison only when the input condition changes from OFF to ON, use the differentiate instruction in conjunction with the input condition.

Instruction	
STR	4003
F-44	
F-12	
	9000
	9010

REFERENCE: To compare data of more than two bytes, it should be programmed like addition (F-10) and subtraction (F-11); the comparison should take place from a lower order digit. If the F-12 were programmed repeatedly, the contents of the carry flag (7356) are also compared after the second F-12 instruction. (For the F-12 instruction that follows the STR instruction, the contents of the carry flag (7356) are removed from the comparison.)



Instruction	
STR	4003
F-12	
	9000
	9010
F-12	
	9001
	9011

When programmed from a low order digit, the carry-down information is forwarded to a higher digit.

Fc12	Compare register with constant
CMP	(CoMPare)

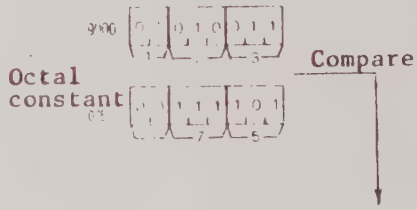
Symbol	
Function	The contents of register S_1 are compared with an octal constant "n".
Operation	S_1 <=> n
Range of S_1 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of "n" applicable	000 ~ 377
Operation condition	When the input signal is in the ON state (not limited to the OFF to ON) transition.
Contents of S_1	Unchanged

Flag	Result	a	b	c	d
	S_1> n	1	0	0	0
	S_1= n	1	0	0	1
	S_1<n	0	0	1	0

a:Non-carry 7354
b:Error 7355
c:Carry 7356
d:Zero 7357

[Explanation]

When the input condition 4001 is ON, the contents of register 9000 are compared with the octal constant 075. Its results are set in the non-carry flag (7354), carry flag (7356), and zero flag (7357). The contents of register 9000 remain unchanged after this operation. It has the timing similar to the F-12 instruction.



Non-carry	Error	Carry	Zero
7354	7355	7356	7356
1	0	0	0

NOTE-1:].734 ~].737 is the special area.

NOTE-2: So long as the input signal is ON, comparison is done on every scan.

NOTE-3: The error flag (7355) is 0 at all times.

NOTE-4: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

REFERENCE: Use an octal number when using the Fc12 instruction. Octal numbers can express any bit pattern and it does not need to use troublesome calculations. To compare with a BCD constant, convert it to an octal equivalent before writing it in the program.

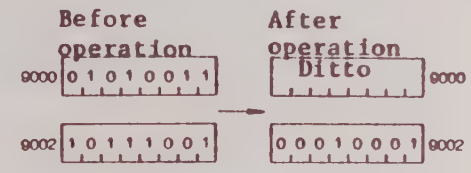
F-13 AND	AND registers (AND)

Symbol	F-13 AND | S | D
Function	The contents of register S (8-bit data) are ANDed with the contents of register D (8-bit data) and its result is stored in register D.
Operation	S ∩ D→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Result
Flag	Unchanged

[Explanation]

Instr.	
STR	4002
F-13	
	9000
	9002

When the input condition 4002 changes from OFF to ON, the 8-bit contents of register 9000 are ANDed with the 8 bit contents of register 9002 and its result is stored in register 9002. The contents of register 9000 remains unchanged.



AND truth table

Symbol	A	B	C
A, B → C	0	0	0
	1	0	0
	0	1	0
	1	1	1

NOTE-1:].734 ~].737 is the special area.

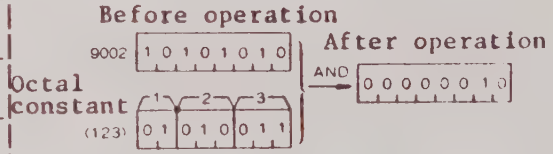
Fc13 AND	AND registers with constant (AND)

Symbol	Fc13 AND | n | D
Function	The octal constant "n" is ANDed with the contents of register D; and its result is stored in register D.
Operation	n ∩ D→D
Range of "n" applicable	000 ~ 377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	Result
Flag	Unchanged

[Explanation]

Instr.	
STR	4001
Fc13	
	123
	9002

When the input condition 4001 changes from OFF to ON, the octal constant 123 is ANDed with the contents of register 9002. Its result is stored in the register 9002.



AND truth table

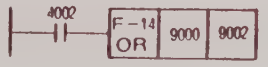
Symbol	A	B	C
A, B → C	0	0	0
	1	0	0
	0	1	0
	1	1	1

NOTE-1:].734 ~].737 is the special area.

F-14 OR	OR registers (OR)

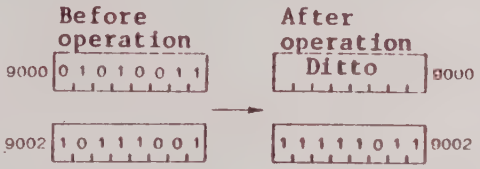
Symbol	F-14 OR | S | D
Function	The contents of register S (8-bit data) are ORed with the contents of register D (8-bit data) and its result is stored in register D.
Operation	S ∪ D→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Result
Flag	Unchanged

[Explanation]



Instr.	
STR	4002
F-14	
	9000
	9002

When the input condition 4002 changes from OFF to ON, the 8-bit contents of register 9000 are ORed with the 8 bit contents of register 9002 and its result is stored in the register 9002. The contents of the register 9000 remains unchanged.



OR truth table

Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

NOTE-1:].734 ~].737 is the special area.

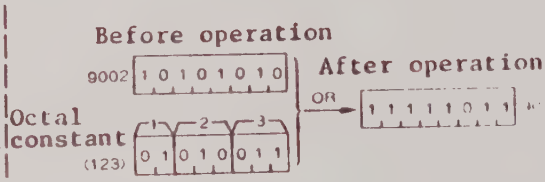
Fc14 OR	OR registers with constant (OR)

Symbol	
Function	The octal constant "n" is ORed with the contents of register D; and its result is stored in register D.
Operation	n ∪ D→D
Range of "n" applicable	000 ~ 377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	Result
Flag	Unchanged

[Explanation]

Instr.	
STR	4001
Fc14	
	123
	9002

When the input condition 4001 changes from OFF to ON, the octal constant 123 is ORed with the contents of register 9002. Its result is stored in the register 9002.



OR truth table

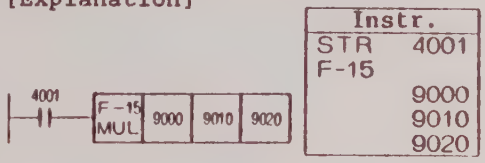
Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

NOTE-1:].734 ~].737 is the special area.

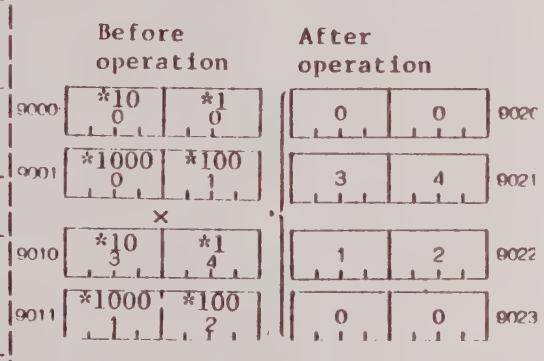
F-15 MUL	Multiply register by register (4-digit) (MULtiply)

Symbol	F-15 MUL | S_1 | S_2 | D
Function	The contents of the register S_1 and S_1+1 (BCD 4 digits) are multiplied by the contents of registers S_2 and S_2+1 (BCD 4 digits). Its result is stored in the 4 byte area starting from register D.
Operation	$(S_1,S_1+1)\times(S_2,S_2+1)$ →D,D+1,D+2,D+3
Range of S_1 Applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of S_2 applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1, S_1+1	Unchanged
Contents of S_2, S_2+1	Unchanged
Contents of D	Results (value in ones and tens order)
Contents of D+1	Results (value in hundreds and thousands orders)
Contents of D+2	Result (value in ten thousands and hundred thousands orders)
Contents of D+3	Result (value in millions and ten millions orders)
Flag	S_1,S_1+1,S_2,S_2+2 | a | b | c | d BCD code | 0 | 0 | 0 | 0 Not BCD code | | 1 | | a:Non-carry 7354 b:Error 7355 c:Carry 7356 d:Zero 7357

[Explanation]



When the input condition 4001 changes from OFF to ON, the BCD 4 digits in registers 9000 and 9001 are multiplied by the BCD 4 digits in registers 9010 and 9011. Its result is stored in the 4-byte area starting from the register 9020.



° The above example shows the operation of 100 x 1234 = 123400.

NOTE-1:].734 ~].737 is the special area.

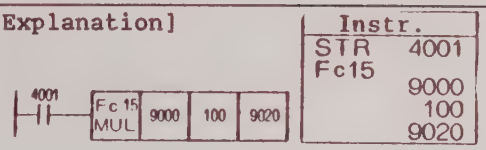
NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

NOTE-3: If the contents of S_1, S_1+1, S_2, or S_2+1 are numbers other than BCD, the error flag (7335) is activated (1) and no multiplication is done. (The contents of D through D+3 remain unchanged.)

Fc15 MUL	Multiply 4-digit register by 3-digit BCD constant (MULtiply)

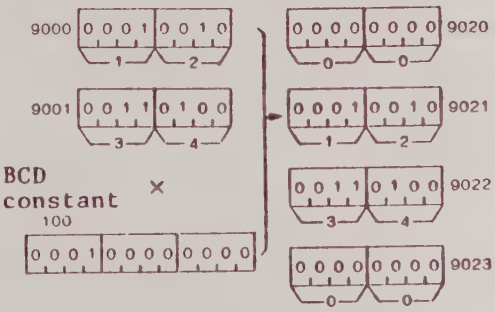
Symbol	
Function	The contents of the register S_1 and S_1+1 (BCD 4 digits) are multiplied by a 3-digit BCD constant "n". Its result is stored in the 4-byte area starting from register D.
Operation	(S_1,S_1+1)xn →D,D+1,D+2,D+3
Range of S_1 Applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of "n" applicable	000~999
Range of D applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1, S_1+1	Unchanged
Contents of D	Results (value in ones and tens order)
Contents of D+1	Results (value in hundreds and thousands orders)
Contents of D+2	Result (value in ten thousands and hundred thousands orders)
Contents of D+3	Result (value in millions and ten millions orders)
Flag	S_1,S_1+1 | a | b | c | d | BCD code | 0 | 0 | 0 | 0 | Not BCD code | | 1 | | | a:Non-carry 7354 b:Error 7355 c:Carry 7356 d:Zero 7357

[Explanation]



When the input condition 4001 changes from OFF to ON, the BCD 4 digits in registers 9000 and 9001 are multiplied by the 3-digit BCD constant 100. Its result is stored in the 4-byte area starting from the register 9020.

3412×100=341200



NOTE-1:].734 ~].737 is the special area.

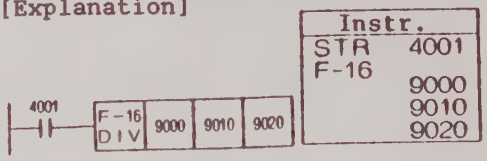
NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

NOTE-3: If the contents of S_1 or S_1+1 are numbers other than BCD, the error flag (7335) is activated (1) and no multiplication is done. (The contents of D through D+3 remain unchanged.)

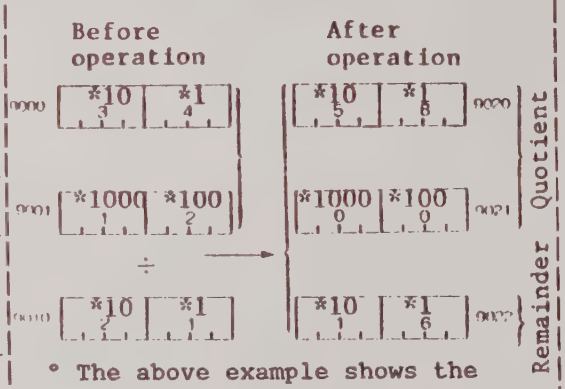
F-16 DIV	Divide 4-digit register by 2-digit register (DIVide)

Symbol	
Function	The contents of the registers S_1 and S_1+1 (BCD 4 digits) are divided by the contents of registers S_2 and S_2+1 (BCD 4-digits). The result is stored in the 2-byte area starting from the register D and the remainder is stored in the third byte area.
Operation	$(S_1, S_1+1)/S_2$ →D, D+1, D+2
Range of S_1 Applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of S_2 applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1, S_1+1	Unchanged
Contents of S_2, S_2+1	Unchanged
Contents of D	Quotient (value in ones and tens order)
Contents of D+1	Results (value in hundreds and thousands orders)
Contents of D+2	Quotient (value in ten thousands and hundred thousands orders)
Flag	S_1, S_1+1, S_2 | a | b | c | d | BCD code | | 0 | | | *Not BCD code | 0 | _ | 0 | 0 | *If S_2 is 00 | 1 | | | | a:Non-carry 7354 b:Error 7355 c:Carry 7356 d:Zero 7357

[Explanation]



When the input condition 4001 changes from OFF to ON, the BCD 4 digits in registers 9000 and 9001 are divided by the BCD 2 digits in registers 9010 and 9011. The quotient is stored in the 2-byte area starting from register 9020 and the remainder is stored in the third byte area.



* The above example shows the operation of 1234 divided by 21 equals 58 with the remainder of 16.

(+):Unchanged if the contents of registers, S_1, S_1+1, or S_2 are not BCD code or the contents of S_2 is 00.

NOTE-1:].734 ~].737 is the special ara.

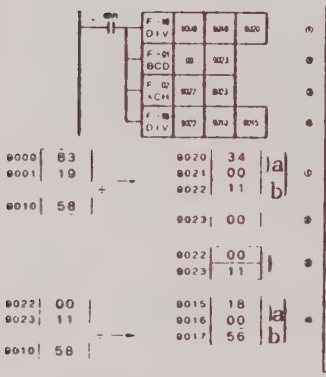
NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

NOTE-3: If the contents of S_1, S_1+1, or S_2 are numbers other than BCD, the error flag (7335) is activated (1) and no division is done. (The contents of D, D+1, and D+2 remain unchanged.)

NOTE-4: If the numberator is smaller than the denominator (S_1 smaller than S_2, S_1+1 = 0), the quotient (D, D+1) will be 0 with the remainder (D+2) will be the numerator (S_1). For instance, execution of 20 divided by 30 produces the result of 0 with the remainder of 20.

REFERENCE: To obtain the result two digit under the decimal point, the following programming is suggested.

(Ex): 1983 / 58 = 34.18 .. remainder 0.56



a:Quotient
b:Remainder

Instruction	
STR	4001
F-16	
	9000
	9010
	9020
F-01	
	00
	9023
F-02	
	9022
	9023
F-16	
	9022
	9010
	9015

1 When the input condition 4001 changes from OFF to ON, the contents of registers 9000 and 9001 are divided by the contents of the register 9010. The quotient is stored in 9020 and 9021 and the remainder is stored in 9022.

2 Enter data 00 to 9023.

3 Exchange the contents of 9022 with 9023 and convert the remainder into the thousands and hundreds orders.

4 Divide the data in 3 by the contents of 9010 again, and store the quotient in 9015 and 9016 with the remainder in 9017. The result under two digits of decimal place is then stored in 9015.

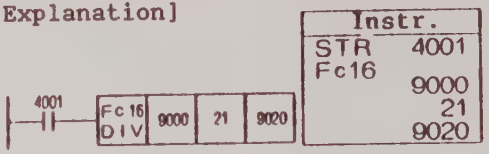
Fc16 DIV	Divide 4-digit register by 2-digit BCD constant (DIVide)

Symbol	
Function	The contents of the registers S_1 and S_1+1 (BCD 4 digits) are divided by a 2-digit BCD constant "n". The result is stored in the 2-byte area starting from the register D and the remainder is stored in the third byte area.
Operation	$(S_1, S_1+1)/n$ →D, D+1, D+2
Range of S_1 Applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Range of "n" applicable	00 ~ 99
Range of D applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S_1, S_1+1	Unchanged
Contents of D	Quotient (value in ones and tens order)
Contents of D+1	Quotient (value in hundreds and thousands order)
Contents of D+2	Remainder

Flag	S_1, S_1+1, S_2	a	b	c	d
	BCD code		0		
	*Not BCD code	0		0	0
	*If n is 00		1		

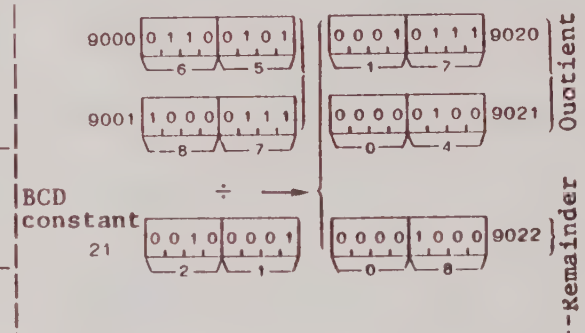
a: Non-carry 7354
b: Error 7355
c: Carry 7356
d: Zero 7357

[Explanation]



When the input condition 4001 changes from OFF to ON, the BCD 4 digits in registers 9000 and 9001 are divided by the BCD constant 21. The quotient is stored in the 2-byte area starting from the register 9020 and the remainder is stored in the third byte area.

8765÷21=417·····8



(+): Unchanged if the contents of registers, S_1 or S_1+1 are not BCD code or the contents of n is 00.

NOTE-1:].734 ~].737 is the special area.

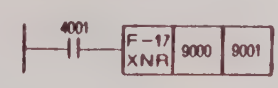
NOTE-2: If the contents of S_1 or S_1+1 are numbers other than BCD, the error flag (7335) is activated (1) and no division is done. (The contents of D, D+1, and D+2 remain unchanged.)

NOTE-3: If the numerator is smaller than the denominator (S_1 smaller than n, S_1+1 = 0), the quotient (D, D+1) will be 0 with the remainder (D+2) will be the numerator (S_1). For instance, execution of 20 divided by 30 produces the result of 0 with the remainder of 20.

F-17 XNR	Exclusive NOR registers (eXclusive NOR)

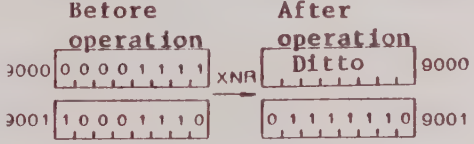
Symbol	F-17 XNR | S | D
Function	The contents of the register S are exclusive NORed with the contents of register D and its result is stored in register D.
Operation	S + D→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Result
Flag	Unchanged

[Explanation]



Instr.	
STR	4001
F-17	9000
	9001

When the input condition 4001 changes from OFF to ON, the contents of register 9000 are exclusive NORed with the contents of register 9001. Its result is stored in register 9001. The contents of the register 9000 remain unchanged.



Bits matched in 9000 and 9001 (0 for 0, 1 for 1) are changed to a 1 and unmatched bits (0 to 1) are changed to a 0.

Exclusive NOR truth table

Symbol	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

NOTE-1:].734 ~].737 is the special area.

Fc17 XNR	Exclusive NOR registers with constant (eXclusive NOR)

Symbol	
Function	The octal constant "n" is exclusive NORed with contents of register D, and its result is stored in the register D.
Operation	S + D→D
Range of "n" applicable	000~377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of D	Result
Flag	Unchanged

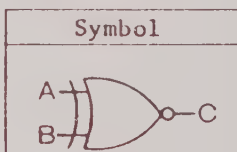
[Explanation]

Instr.	
STR	4001
Fc17 XNR	017
	9001

When the input condition 4001 changes from OFF to ON, the octal constant 017 is exclusive NORed with the contents of register 9001. Its result is stored in register 9001.

Before operation

Exclusive NOR truth table



Symbol	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

NOTE-1:].734 ~].737 is the special area.

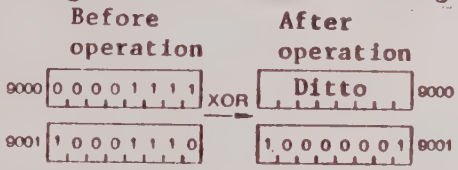
F-18 XOR	Exclusive OR registers (eXclusive OR)

Symbol	
Function	The contents of the register S are exclusive ORed with the contents of register D and its result is stored in register D.
Operation	S + D→D
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Result
Flag	Unchanged

[Explanation]

Instr.	
STR	4001
F-18	
	9000
	9001

When the input condition 4001 changes from OFF to ON, the contents of register 9000 are exclusive ORed with the contents of register 9001. Its result is stored in register 9001. The contents of the register 9000 remain unchanged.



Bits matched in 9000 and 9001 (0 for 0, 1 for 1) are changed to a 0 and unmatched bits (0 to 1) are changed to a 1.

Exclusive ORed truth table

Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

NOTE-1:].734 ~].737 is the special area.

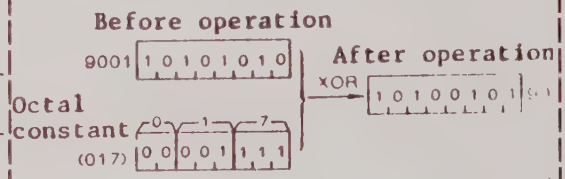
Fc18 XOR	Exclusive OR registers with constant (eXclusive OR)

Symbol	
Function	The octal constant "n" is exclusive ORed with contents of register D, and its result is stored in the register D.
Operation	S + D→D
Range of "n" applicable	000~377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of signal (OFF to ON)
Contents of S	Unchanged
Contents of D	Result
Flag	Unchanged

[Explanation]

Instr.	
STR	4001
Fc18	
	017
	9001

When the input condition 4001 changes from OFF to ON, the octal constant 017 is exclusive ORed with the contents of register 9001. Its result is stored in register 9001.



Exclusive OR truth table

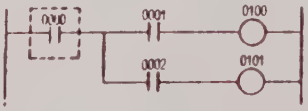
Symbol	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

NOTE-1:].734 ~].737 is the special area.

| F-30 | Set master control
| MCS | (Master Control Set)

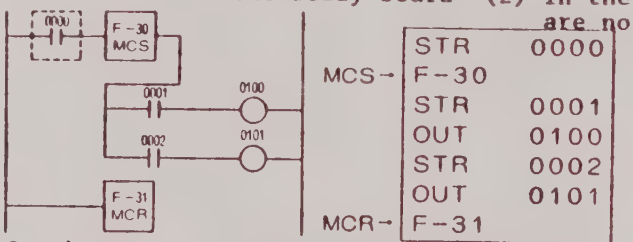
| F-31 | Reset master control
| MCR | (Master Control Reset)

MCS and MCR should be used when the circuit after the common operational condition is branched to multiple outputs.



(1) In the case of the relay board

(2) In the case of MCS and MCR are not used



(3) In the case of MCS and MCR are used

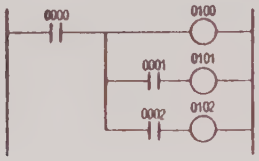
When the F-30 (MCS) instruction is programmed, the contents of the accumulator is stored in the CPU internal register, and operations for successive instructions are ANDed with the contents of the CPU internal register until the F-31 (MCR) instruction is reached. The F-31 (MCR) instruction indicates The range of AND operation applied.

It will help simplify the program when the command operational condition is indicated in the block |____| is compiled or many branches are set after the common operational condition.

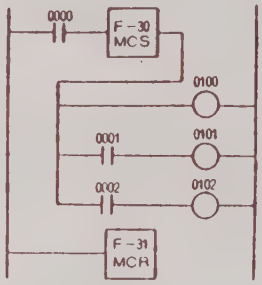
The MCS instruction turns OFF associated contacts, coils, timers, and counters. However, a register will not be cleared; and the outputs associated with (driven by) it will retain their last state. A register should not reside within the MCS and MCR commands.

NOTE-1: Do not directly connect the F-30 (MCS) derived bus line with OUT, TMR, and CNT instrucitons and applicaiton instruction.

(1) In case of the relay board

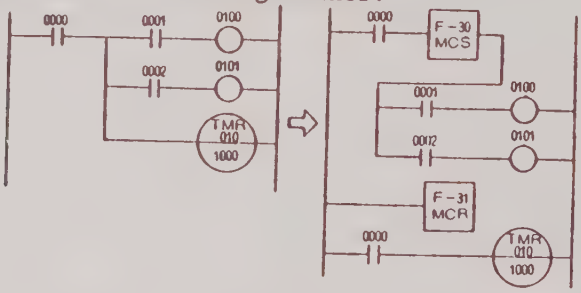


(2) MCS, MCR prohibited program



Instruction	
STR	0000
F-30	
OUT	0100
STR	0001
OUT	0101
STR	0002
OUT	0102
F-31	

Program in the following manner:

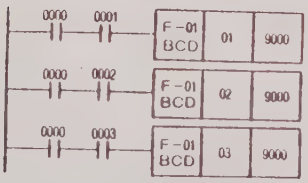


NOTE-2: The F-31 (MCR) instruction is an unconditional instruction.

Incorrect use

Correct use

NOTE-3: OFF to ON condition of the F-30 (MCS) instruction cannot be used for the operational condition of the CNT, or application instructions which execute the operation at a OFF to ON transition.

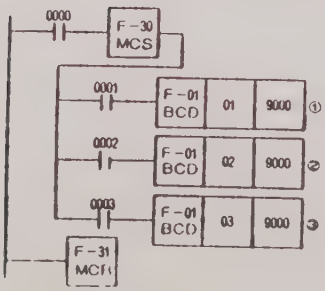


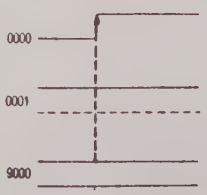
Program designed to transfer data to 9000 when 0000 changes from OFF to ON;

the BCD constant 01, if 0001 is ON,
the BCD constant 02, if 0002 is ON,
the BCD constant 03, if 0003 is ON.

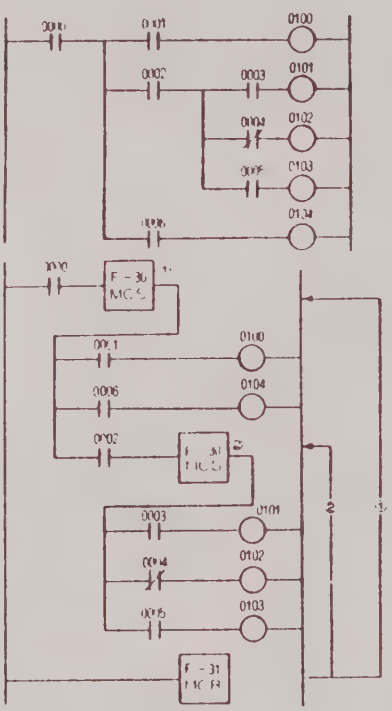
0000 ... Pushbutton switch contact
0001~0003 ... Rotary switch contacts

Operation would not take place if the above program is written in the following manner using MCS and MCR.





Another MCS may be used in-between MCS and MCR.

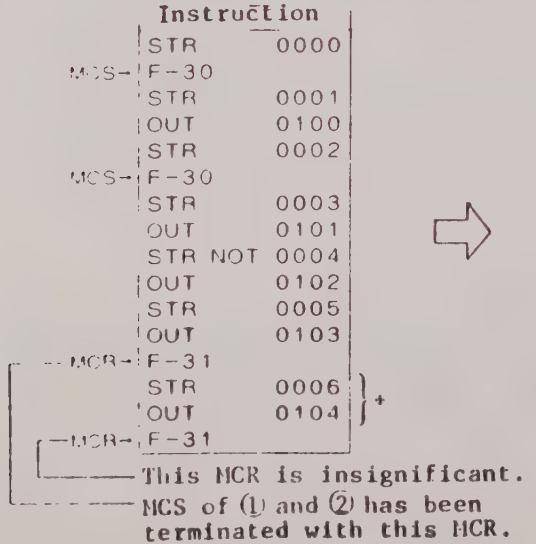


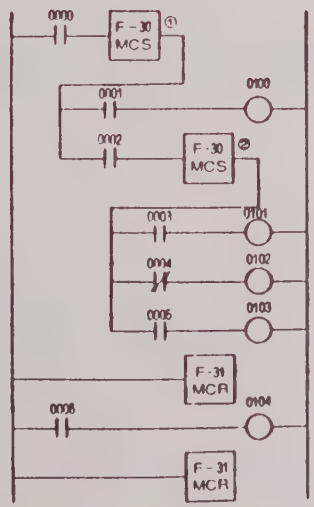
Relay board ladder chart shown (left) can be programmed in the following manner using MCS and MCR. However, there may be a need to change the program sequence. (*)

INSTRUCTION	
STR	0000
F-30	
STR	0001
OUT	0100
STR	0006 *
OUT	0104 *
STR	0002 *
F-30	
STR	0003
OUT	0101
STR NOT	0004
OUT	0102
STR	0005
OUT	0103
F-31	

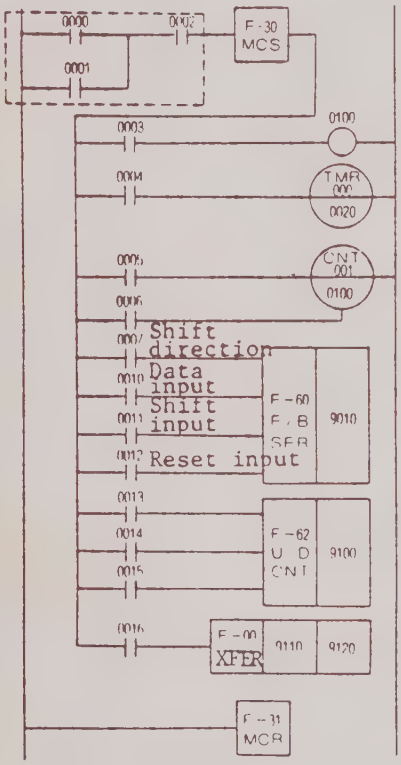
° The F-31 (MCR) instruction indicates the termination of the preceding F-30 (MCS) instruction: 1 and 2 in the example.

NOTE-3: The MCR's reset function:





When the MCS condition (shown within dotted line) is OFF, instructions existing between MCS and MCR will be handled in the following manner.



*MCS 0003 — 0100 is OFF because it is OFF.

*MCS 0004 — TMR is reset because it is OFF. (TMR contact turns to OFF and the current value to preset value.)

*MCS 0005, 0006 — Because it is OFF;

	ON reset	OFF reset
CNT contact	Retained	OFF
Current value	Retained	Preset value

*MCS 0007, 0010, 0011, 0012 — Because it is OFF,

	ON reset	OFF reset
9010	Retained	Cleared
Flag	Cleared	Cleared

*MCS 0013, 0014, 0015 — Because it is OFF;

	ON reset	OFF reset
9100	Retained	Cleared
Flag	Cleared	Cleared

*MCS 0016 — No operation is executed because it is OFF.

*MCS ... MCS condition

	Flag
Flag-unaffecting instruction	No change
Flag-affecting instruction	Cleared

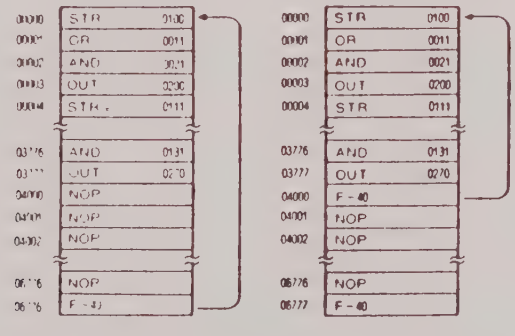
NOTE-4: CNT, F-60, and F-62 instructions can set the reset condition to ON-reset or OFF-reset in the system memory #202. In the case of OFF-reset, it is reset by MCS.

| F-40 | End instruction
| END | (END)

The F-40 instruction indicates the end of the program. As it has the highest priority, it will be effective even if it's between F-41 (JCS) and F-42 (JCR) (it terminates the user program). There is no need of writing the END instruction, except for the following two cases; because it will be automatically set in the last address of the program memory when the memory is cleared after specifying the program memory capacity.

(1) To accelerate the scan time

The scan time is the total I/O processing time plus the user program execution time. The user program execution time is the total time required to execute all instructions from the program address 00000 to the END instruction. The location of the END instruction is automatically written after memory clear is. For instance, 06777 (3584th byte) for 3.5KW. Assume that the last program address is 03777 (2048th word); 04000 ~ 06777 are padded with NOP instructions and the END instruction is in 06777. Time has to be consumed for execution of all of the NOP instructions (1.21 microsecond per word) between 04000 and 04776. If F-40 is written in 04000, it accelerates the processing time because it does not execute those unwanted NOP instructions and the control can proceed to a next scan cycle.



(2) To perform partial program execution during trial run by inserting the F-40 instruction at the end of a sequence block, you will be able to execute a specific portion of the program. If the result was successful, that F-40 is deleted.

NOTE-1: When the zero cross synchronization mode is used, the scan time will be dependent on the zero cross synchronization.

NOTE-2: A multiple number of F-40's may exist when memory is added or the END instruction is written in (2). In this case, the user program terminates at the first F-40. Check the location of the END instruction before going into the actual operation.

NOTE-3: The F-40 (END) instruction is unconditional.

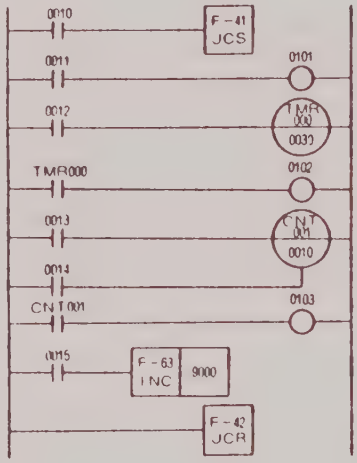
Incorrect

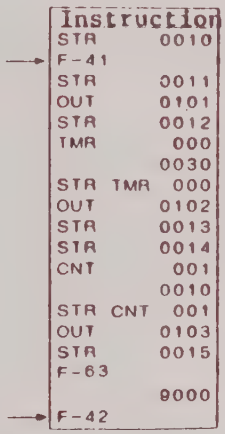
Correct

| F-41 | Set jump control
| JCS | (Jump Control Set)

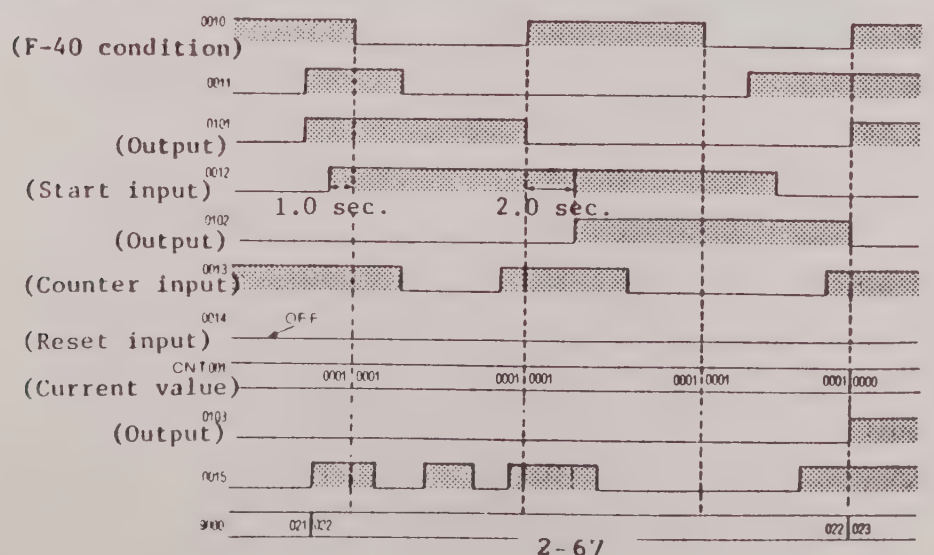
| F-42 | Reset jump control
| JCR | (Jump Control Reset)

When the F-41 (JCS) condition is OFF, all instructions (except the END instruction) before the F-42 (JCR) instruction are not executed. If there was an instruction whose result should be stored in the data memory, it will not change the data memory contents. The data memory retains the state at the time of JCS condition is ON.





NOTE-1: Attention must be paid to the TMR internal clock (0.1-second clock), CNT counter input and application instruction input condition (OFF to ON transition in the input condition), and F-41 (JCS) condition ON/OFF timing.



Operation takes place at the rising of 1 because the JCS condition is ON.

Operation does not take place at the rising of 2 because the JCS condition is OFF.

Operation does not take place at a rising of 3 because the JCS condition is OFF.

The JCS condition turns to ON while 3 is ON, but operation does not take place as it does not recognize that the input condition has changed from OFF to ON.

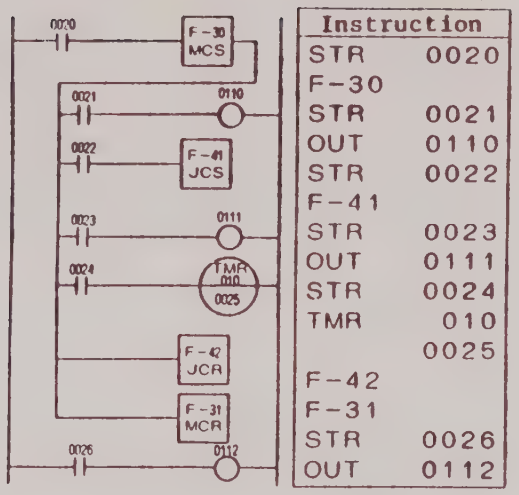
Operation does not take place at a rising of 4 because the JCS condition is OFF.

The JCS condition becomes ON while 4 is ON.

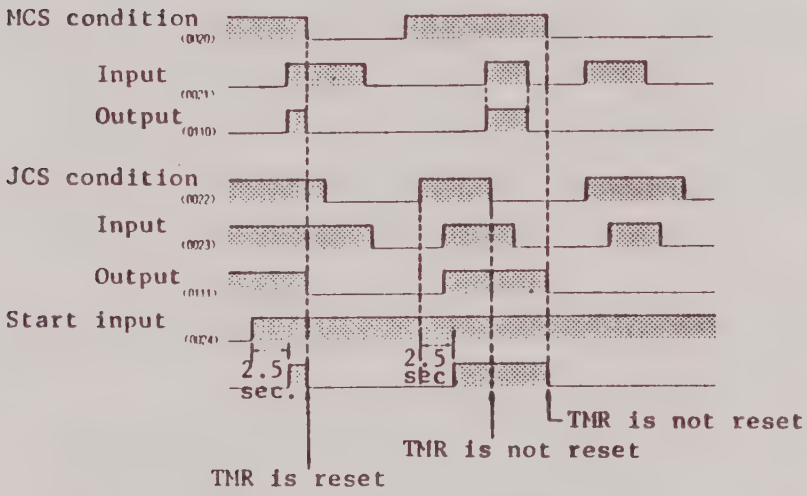
NOTE-2: Operation does not take place if the JCS condition is OFF when there is a flag-affecting instruction between F-41 (JCS) and F-42 (JCR). Instead, flags 7354 through 7357, are cleared.

NOTE-3: The END instruction will be executed whether the JCS is ON or OFF; if there was a F-40 (END) instruction between F-41 (JCS) and F-42 (JCR), and if the user program execution is terminated and the control proceeds to a next scan cycle.

NOTE-4: F-41 (JCS) and F-42 (JCR) can be nested between F-30 (MCS) and F-31 (MCR). However, as MCS has a higher priority than JCS, it assumes the state of the MCS.

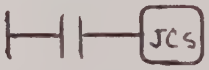


Instruction	
STR	0020
F-30	
STR	0021
OUT	0110
STR	0022
F-41	
STR	0023
OUT	0111
STR	0024
TMR	010
	0025
F-42	
F-31	
STR	0026
OUT	0112



NOTE-5: It is not permitted to insert F-41 or F-42 between F-41 (JCS) and F-42 (JCR). "JCS ERROR" will be indicated on the P110 programmer during program check.

NOTE-6: The following kind of programming is not possible as F-42 (JCR) is an unconditional instruction.



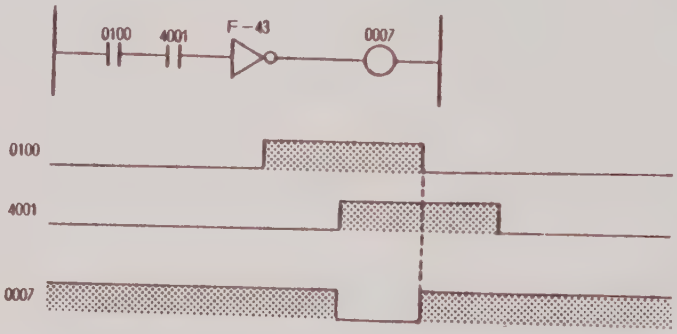
Incorrect

Correct

F-43	Complement bit (invert)
CPL	(ComPLement)

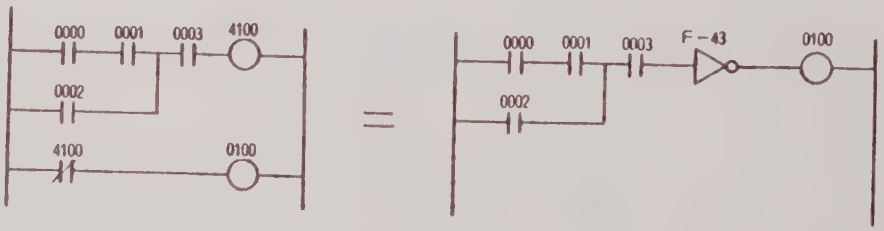
The F-43 instruction complements the bit in the accumulator.

Instruction	
STR	0100
AND	4001
F-43	
OUT	0007



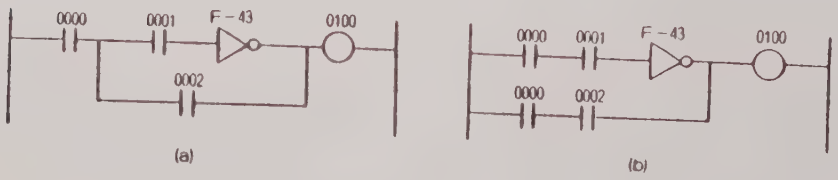
The result of the STR and instruction is complemented by the F-43 instruction and set to output relay 0007.

Use of F-43 eliminates the use of the auxiliary contact, as shown below.



NOTE-1: The F-43 instruction may be used for any number of contacts.

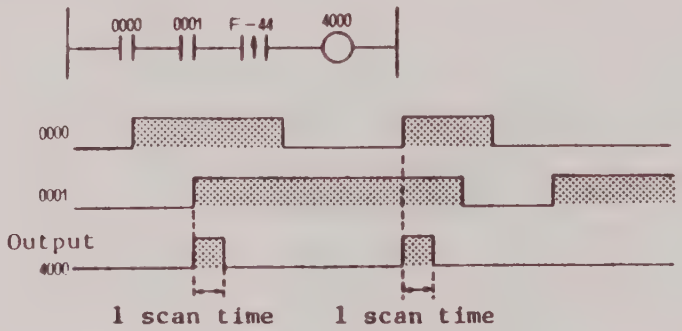
NOTE-2: Because the F-43 instruction complements the content of the accumulator, these programs are not the same.



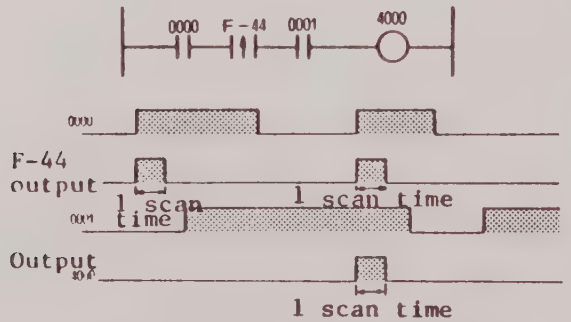
| F-44 | Differentiate at ON
|-|↑|-|

When the state of the accumulator just before the F-44 instruction changes from OFF to ON, one scan time pulse is generated.

Instruction	
STR	0000
AND	0001
F-44	
OUT	4000

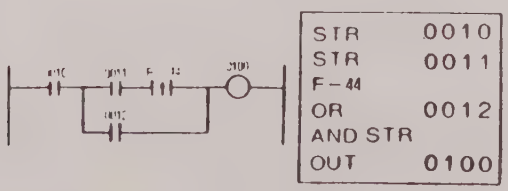


NOTE-1: Note that a different result is produced when the sequence of the F-44 program is changed from the above ladder chart. (Identical in the case of F-45.)

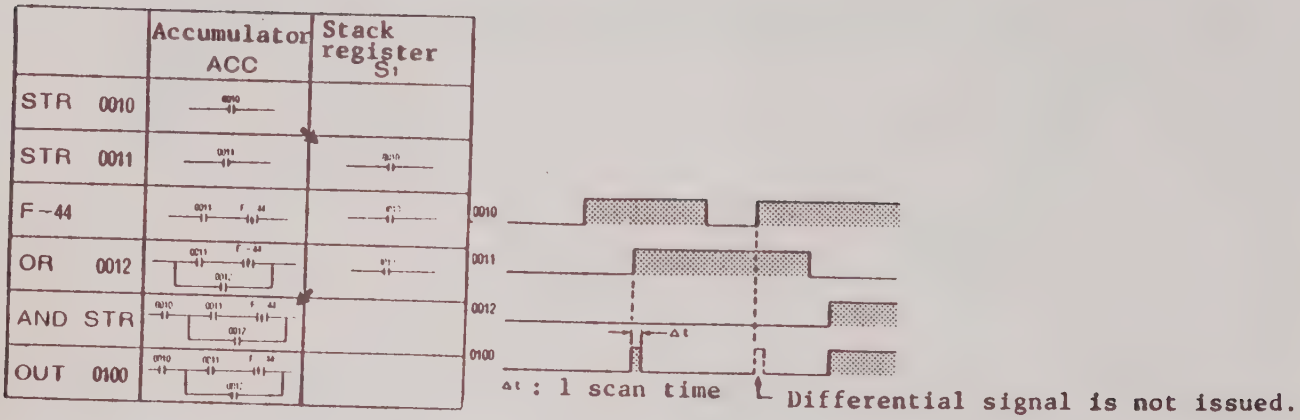


NOTE-2: F-44 may use any number of contacts.

NOTE-3: F-44 is the instruction that produces a signal that turns ON for one scan by detecting an OFF to ON change in the contents of the accumulator.



STR	0010
STR	0011
F-44	
OR	0012
AND STR	
OUT	0100

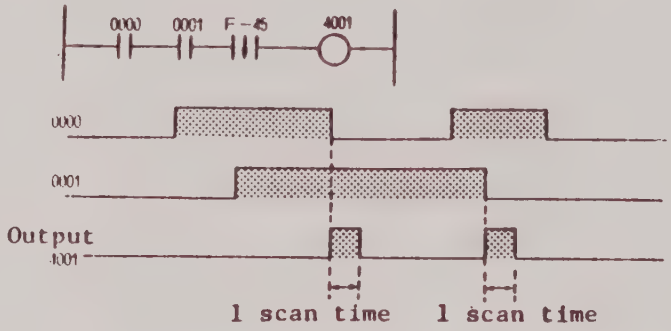


In the above example, a differential signal is not issued even if 0010 has changed from OFF to ON when 0011 is ON, because 0010 is ANDed by the AND STR instruction.

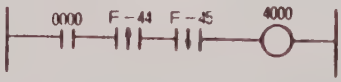
| F-45 | Differentiate at OFF
|-|↓|-|

When the state of the accumulator just befor the F-45 instruction changes from ON to OFF, one scan time pulse is generated.

Instruction	
STR	0000
AND	0001
F-45	
OUT	4001

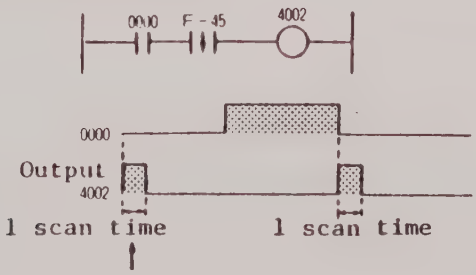


NOTE-1: This ladder logic will not generate the desired pulse.



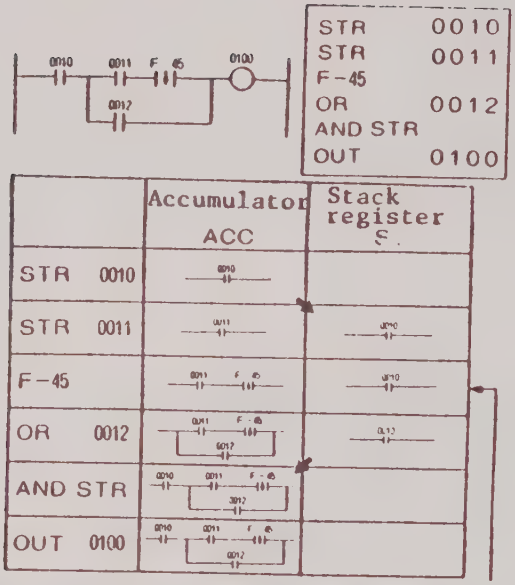
NOTE-2: You may use differential instructions (F-44,F-45) as many times as required in your program.

NOTE-3: Use of the F-45 instruction may sometimes operate immediately following program writing.

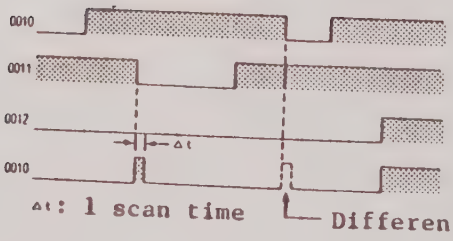


Output (4002) turns ON if input (000) is OFF at the start of operation.

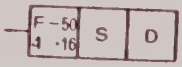
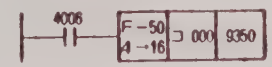
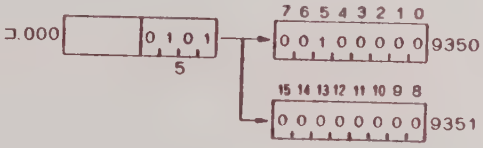
NOTE-4: F-45 produces a signal that turns ON for one scan by detecting an ON to OFF change in the contents of the accumulator.



The accumulator turns ON for a scan when 0011 changed from ON to OFF.

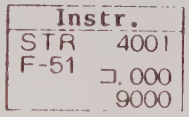


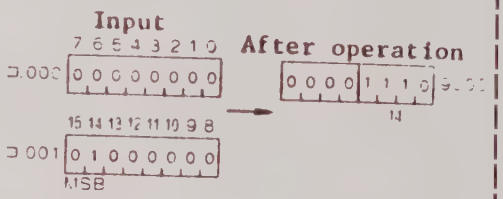
In the above example, differentiate signal is not issued even if 0010 changed from ON to OFF when 0011 is OFF, because 0010 is ANDed by the AND STR instruction.

Symbol			
Function		The low order 4 bits of register S are decoded and stored in the 2-byte area of registers D and D+1.	(Explanation)  Instr. STR 4006 F-50 ⊐.000 9350 When the input condition 4006 changes from OFF to ON, the low order 4 bits of].000 are decoded and stored in the 2-byte area of registers 9350 and 9351. 
Operation		S→D, D+1	
Range of S applicable		].000~].733 (NOTE-1) b.000~b.377 9000~9377	
Range of D applicable		].000~].732 (Note-1) b.000~b.377 9000~9377	
Operational condition		At the rising edge of signal (OFF to ON).	
After the operation	Contents of S	Unchanged	Only the bit position that corresponds to the low order 4 bits representing 0 to 15 is turned ON.
	Contents of D	Result (0~7)	
	Contents of D+1	Result (8~15)	
	Flag	Unchanged	

NOTE-1:].734 ~].737 is the special area.

NOTE-2: High order 4 bits of S are disregarded.

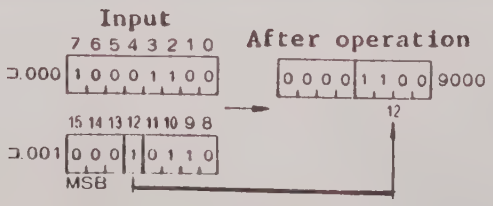
Symbol		
Function	The 2-byte data in registers S and S+1 are encoded and stored in register D.	(Explanation)  When the input condition 4001 changes from OFF to ON, the 2-byte data consisting of registers].000 and].001 are encoded and stored in the register 9000.
Operation	S, S+1→D	
Range of S applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377	
Range of D applicable	].000~].733 (Note-1) b.000~b.377 9000~9377	
Operational condition	At the rising edge of signal (OFF to ON).	
After the operation: Contents of S	Unchanged	
After the operation: Contents of S+1	Unchanged	
After the operation: Contents of D	Result	
After the operation: Flag	Unchanged	



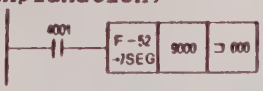
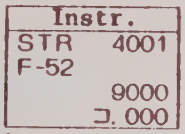
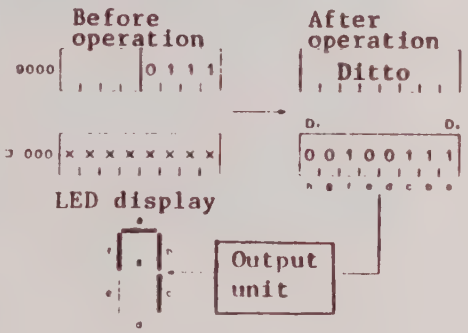
NOTE-1:].734 ~].737 is the special area.

NOTE-2: After the operation, the high order 4 bits of D (9000 in the example) are always 0.

NOTE-3: The MSB side of the encoder input takes the highest priority.



F-52 →7SEG	Decode to 7-segment data

Symbol		
Function	The low order 4-bit data in register S are decoded into 7-segment display data.	(Explanation)   When the input condition 4001 changes from OFF to ON, the low order 4 bits of the register 9000 are decoded into 7-segment display data. 
Operation	S→D	
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377	
Range of D applicable	].000~].733 (Note-1) b.000~b.377 9000~9377	
Operational condition	At the rising edge of signal (OFF to ON).	
After the operation — Contents of S	Unchanged	
After the operation — Contents of D	Result	
After the operation — Flag	Unchanged	

NOTE-1:].734 ~].737 is the special area.

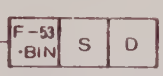
NOTE-2: D_0 ~ D_6 of the output data correspond to a ~ g of the 7-segment display. D_7 is "0" at all times.

7-segment decoder chart

Input data	Output data	Display output
00000000	00111111	0
00000001	00000110	1
00000010	01011011	2
00000011	01001111	3
00000100	01100110	4
00000101	01101101	5
00000110	01111101	6
00000111	00100111	7
00001000	01111111	8
00001001	01101111	9
00001010	01110111	a
00001011	01111100	b
00001100	00111001	c
00001101	01011110	d
00001110	01111001	e
00001111	01110001	f

F-53 →BIN Convert 4-digit BCD to 16-bit binary

Symbol



Function

Four digit BCD data (two bytes) in registers S and S+1, are converted into binary and and stored in two bytes in registers D and D+1.

Operation

S,S+1→D,D+1

Range of S applicable

].000~].732 (NOTE-1)
b.000~b.377
9000~9377

Range of D applicable

].000~].732 (Note-1)
b.000~b.377
9000~9377

Operational condition

At the rising edge of signal (OFF to ON).

After the operation

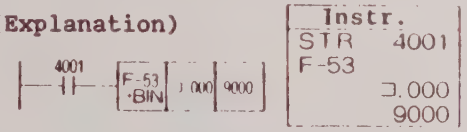
Contents of S, S+1	Unchanged	
Contents of D	Result (0~255)	Not changed if the contents of registers, S or S+1, are not BCD code.
Contents of D+1	Result (256~9999)	

Flag

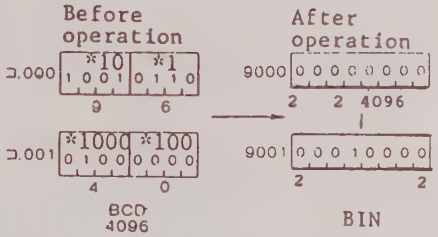
S, S+1	a	b	c	d
BCD code		0		
Not BCD code	0	1	0	0

a: Non-carry 7354
b: Error 7355
c: Carry 7356
d: Zero 7357

(Explanation)



When the input condition 4001 changes from OFF to ON, the contents of the 4-digit BCD data in registers].000 and].001 are converted into a binary equivalent and stored in two bytes of registers 9000 and 9001.

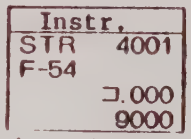
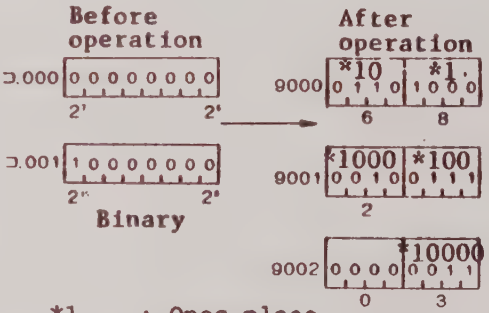


*1 : Ones place
*10 : Tens place
*100 : Hundreds place
*1000: Thousands place

NOTE-1:].734 ~].737 is the special area.

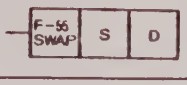
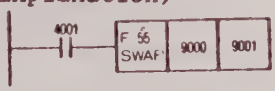
NOTE-2: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

F-54 →BCD Convert 16-bit binary to 6-digit BCD

Symbol		
Function	Two bytes binary data in registers S and S+1, are converted into a BCD 6 digits and stored in three bytes in registers D, D+1, and D+2.	(Explanation)  When the input condition 4001 changes from OFF to ON, the 2-byte contents of binary data in registers].000 and].001, are converted into BCD 6 digits and stored in three bytes beginning from the register 9001.
Operation	S,S+1→D,D+1,D+2	
Range of S applicable	].000~].732 (NOTE-1) b.000~b.377 9000~9377	
Range of D applicable	].000~].731 (Note-1) b.000~b.377 9000~9377	 *1 : Ones place *10 : Tens place *100 : Hundreds place *1000 : Thousands place *10000: Ten thousands place
Operational condition	At the rising edge of signal (OFF to ON).	
After the operation: Contents of S,S+1	Unchanged	
After the operation: Contents of D	Result (ones and tens orders)	
After the operation: Contents of D+1	Result (hundreds and thousands orders)	
After the operation: Contents of D+2	Result (ten thousands order)	
Flag	Unchanged	

NOTE-1: .]734 ~].737 is the special area.

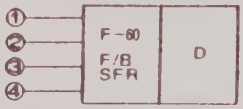
F-55 SWAP (SWAP) Swap high order 4 bits with low order 4 bits

Symbol		
Function	High order 4 bits are swapped with low order 4 bits of register S and stored in register D.	(Explanation)  Instr. STR 4001 F-55 9000 9001 When the input condition 4001 changes from OFF to ON, the high order 4 bits are swapped with low order 4 bits of register 9000. Its result is stored in register 9001. The contents of register 9000, however, remain unchanged. Before operation After operation ×: Don't care
Operation	S→D	
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377	
Range of D applicable	].000~].733 (Note-1) b.000~b.377 9000~9377	
Operational condition	At the rising edge of signal (OFF to ON).	
After the operation: Contents of S	Unchanged	
After the operation: Contents of D	Result	
After the operation: Flag	Unchanged	

NOTE-1: .]734 ~ .]737 is the special area.

NOTE-2: F-55 is useful in the following case:
With the F-52 (7SEG decode) instruction, the low order 4 bits are decoded into 7-segment display data. To make a display with more digits, the high order 4 bits can be swapped with the low order 4 bits. Then F-52 can be executed.

F-60 SFR — Shift register bidirectional (Forward/Backward Shift Register)

Symbol			1) Shift direction input 2) Data input 3) Shift input 4) Reset input			
Function		The 8-bit data in register S is shifted to high order or low order bit positions according to the shift direction input.				
Operation		°Shift direction input is ON: °Shift direction input is OFF: Data input→				
Range of D applicable		.]000~].733(NOTE-1), b.000~b.377, 9000~9377				
Operational condition		When the reset input is OFF, bits are shifted at a rising edge of the shift input.				
After the operation	Contents of D	°Result is contained when the reset input is OFF. °All bits are reset of OFF when the reset input is ON.				
	Flag	Reset input 4)	Non-carry 7354	Error 7355	Carry 7356	Zero 7357
		OFF	1 or 0	0	0 or 1	0 or 1
		ON	0		0	0

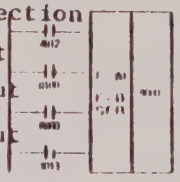
NOTE-1: .]734 ~ .]737 is the special area.

NOTE-2: Use the reset condition in the system memory (#202), to reset with OFF.

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

(Explanation)

Shift direction
Data input
Shift input
Reset input



Instr.
STR 4012
STR 0100
STR 0000
STR 4013
F-60
9000

When shift input 0000 changes from OFF to ON, data are shifted in the following manner depending on the state of the shift direction input 4012.

When 4012 is ON

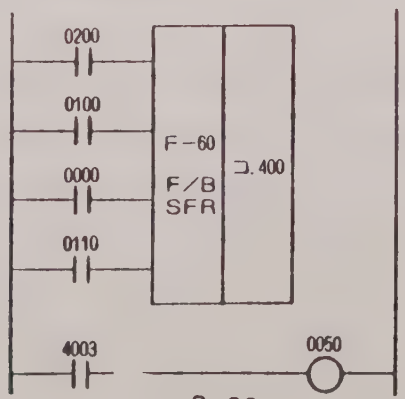
When 4012 is OFF

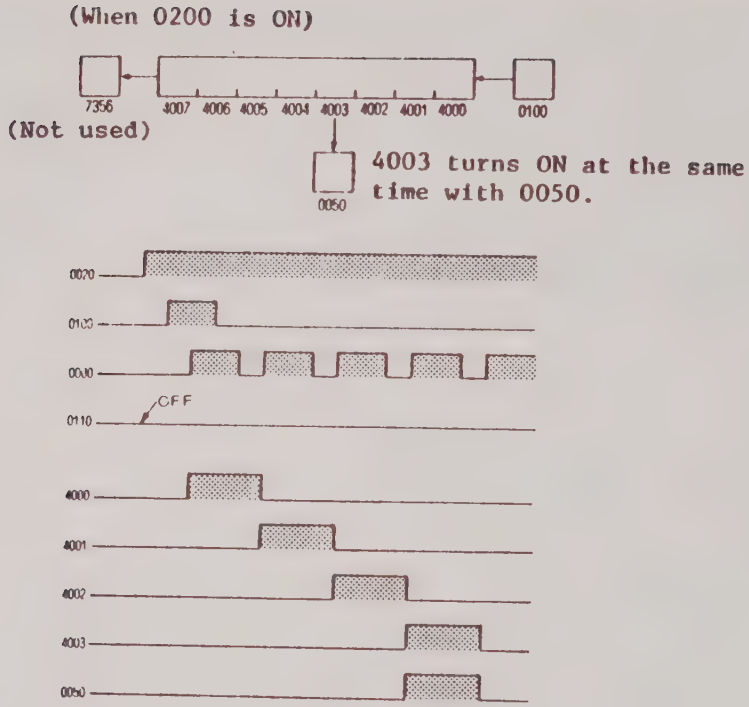
Input condition	9000 (before oper.) 7 6 5 4 3 2 1 0	9000 (after oper.) 7 6 5 4 3 2 1 0	Non-carry 7354	Carry 7356	Zero 7357
4012 O	O O O O O O O O	O O O O O O O O	●	O	●
0100 O	● O O O ● O ● O	O ● O O O ● O ●	●	O	O
0000	O ● O O ● O O ●	O O ● O O ● O O	O	●	O
4013 O	O O O O O O O ●	O O O O O O O O	O	●	●
4012 ●	O O O O O O O O	O O O O O O O O	●	O	●
0100 O	O ● O ● O O O O	● O ● O O O O O	●	O	O
0000	● O O O ● O O O	O O O ● O O O O	O	●	O
4013 O	● O O O O O O O	O O O O O O O O	O	●	●
4012 O					
0100 O	● O O O O ● O O	● ● O O O O ● O	●	O	O
0000					
4013 O	O O O ● O O O ●	● O O O O O O ●	●	O	O
4012 ●					
0100 ●	O ● O O O O O O	● O O O O O O ●	●	O	O
0000					
4013 O	● O O O ● O O O	O O O ● O O O ●	O	●	O
4013 ●	O O ● O ● O O O	O O O O O O O O	O	O	O

° Error Flag (7355) is OFF at all times.

O = OFF ● = ON

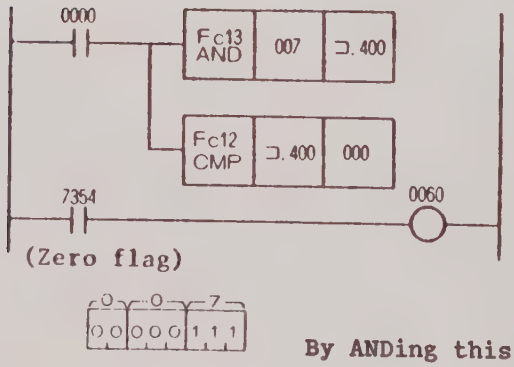
REFERENCE: When .]xxx is assigned for register D, this instruction forms an n-bit (n<8) shift register.





NOTE-1: Data is shifted in 4004 ~ 4007.

NOTE-2: When all of 4000 through 4007 are 0, the zero flag turns to 1. The following program must be used to check if 4000 ~ 4002 are 0.



By ANDing this way, 4003 ~ 4007 are masked (all 0).

F-61 Shift register asynchronous
ASFR (asynchronous Shift Register)

Symbol						1) Shift direction input 2) Shift input	
Function		One byte of data in register D-1 Shift direction input ON) or register D+1 Shift direction input OFF) are shifted to register D.					
Operation		°Shift direction input is ON: D-1 → D °Shift direction input is OFF: D+1 → D					
Range of D applicable		.]001~].732(NOTE-1) b.000~b.377 9000~9377					
Operational condition		When the contents of register is D is 0, shift is done with the ON state of the shift input (not limited to the OFF to ON change).					
		1) ON		1) OFF		1) ON/OFF	
		Before operation	After operation	Before operation	After operation	Before operation	After operation
Register	Contents of D-1	D1	0	D1	Ditto	D1	Ditto
	Contents of D	0	D1	0	D2	Other than 0	Ditto
	Contents of D+1	D2	Ditto	D2	0	D2	Ditto
Flag	Non-carry 7354	1 (D1=0) 0 (D1=0)		1 (D2=0) 0 (D2=0)		1	
	Error 7355	0		0		0	
	Carry 7356	0 (D1=0) 1 (D1=0)		0 (D2=0) 1 (D2=0)		0 0	
	Zero 7357	0		0		0	

NOTE-1:].734 ~].737 is the special area.

NOTE-2: The contents of the shift register (D-1 of D+1) are cleared upon execution.

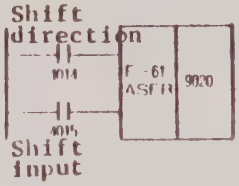
NOTE-3: Execution will not take place unless the contents of D is not 0.

NOTE-4: The carry flag (7356) is set ON olny when data other than 0 is shifted from D-1 or D+1.

NOTE-5: Operation takes place in the scan cycle while the input is ON.

NOTE-6: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

(Explanation)



Instr.
STR 4014
STR 4015
F-61
9020

One byte of data will be shifted in the following way while the shift input 4015 is ON, according state of the shift direction input 4014.

°Shift does not take place unless the contents of 9020 is 0 before the execution.
°The contents of the shifted register (9017 or 9021) are cleared.

When 4014 is ON

When 4014 is OFF

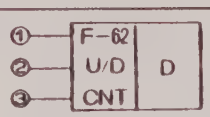
Input condition		Before oper. 7 6 5 4 3 2 1 0	After oper. 7 6 5 4 3 2 1 0	Non-carry 7354	Carry 7356	Zero 7357
4014	●	9017 ○○●○○○○●	○○●○○○○●	●	○	○
4015	●	9020 ○●○○○○●○	○●○○○○●○			
		9021 ●○○○○●○○	●○○○○●○○			
4014	●	9017 ○○○○●○●○	○○○○○○○○	○	●	○
4015	●	9020 ○○○○○○○○	○○○○●○●○			
		9021 ●●○○○○○○	●●○○○○○○			
4014	○	9017 ○○○○●○●○	○○○○●○●○	○	●	○
4015	●	9020 ○○○○○○○○	●●○○○○○○			
		9021 ●●○○○○○○	○○○○○○○○			
4014	○	9017 ○○○○●○●○	○○○○●○●○	●	○	○
4015	●	9020 ○○○○○○○○	○○○○○○○○			
		9021 ○○○○○○○○	○○○○○○○○			

*The error flag (7355) will be OFF at all times.

○ OFF ● ON

F-62	Set up/down counter
U/DCNT	(Up/Down Counter)

Symbol



1) Up/down counter direction input
2) Counter input
3) Reset input

Function

The contents of the (BCD 2 digits) are added (1 is ON) or subtracted (1 is OFF), according to the up/down counter direction input.

Operation

1) Up/down direction input is ON: <D>+1→D
1) Up/down direction input is OFF: <D>-1→D

Range of D applicable

].000~].733 (NOTE-1)
b.000~b.377
9000~9377

Operational condition

At a rising of the counter input (OFF to ON) when the reset input is OFF.

After the operation

Contents of D

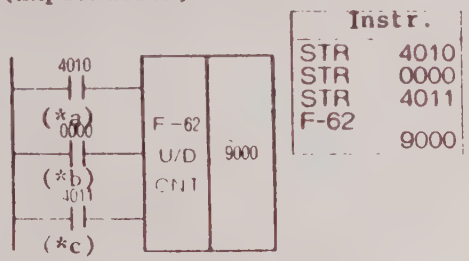
- Result (BCD code) is contained when the reset input is OFF.
- All bits are OFF when reset input is ON

Flag

(+)	Result	a	b	c	d
	99+1→00	0	0	1	1
	00~98+1 →01~99	1	0	0	0
ON	Other than BCD	0	1	0	0
	00-1→99	0	0	1	0
	01-1→00	1	0	0	1
OFF	02~99-1 →01~98	1	0	0	0
	Other than BCD	0	1	0	0
Reset input3)		0	0	0	0

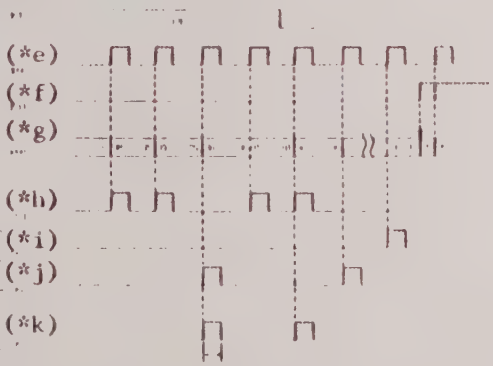
(+) :Up/down counter direction input 1)
a: Non-carry 7354
b: Error 7355
c: Carry 7356
d: Zero 7357

(Explanation)



(*a) :Up/down counter direction
(*b) :Counter input
(*c) :Reset input

Counting is permitted when the reset input 4011 is OFF. When the up/down counter direction input 4010 is ON, it acts as an incremental counter. When OFF, it acts as a decremental counter. If the contents of 9000 were other than BCD code, the error flag (7355) is turned ON and no counter operation takes place. (7F in the example)



1 scan time, max.
Valid until a flag-affecting instruction is met in the program.

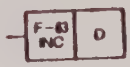
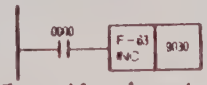
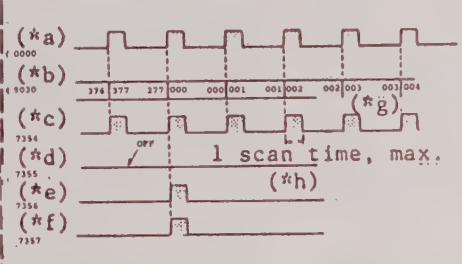
(*d):Up/down counter direction
(*e):Counter input
(*f):Reset input
(*g):Register
(*h):Non-carry flag
(*i):Error flag
(*j):Carry flag
(*k):Zero flag

NOTE-1:].734 ~].737 is the special area.

NOTE-2: By setting the reset condition in the system memory (#202), it is possible to reset with an OFF condition.

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

F-63	Increment counter
INC	(Increment)

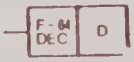
Symbol		
Function	The contents of register D (binary data) are incremented.	(Explanation)  When the input condition 0000 changes from OFF to ON, it increments the data.
Operation	<D>+1→DD	
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377	 (*a): Input (*b): Register value (*c): Non-carry flag (*d): Error flag (*e): Carry flag (*f): Zero flag (*g): 377 (8) indicates an octal number 377. (*h): Valid until a flag affecting instruction is met in the program.
Operational condition	At the rising edge of signal (OFF to ON).	
Contents of D	Reseult (binary code)	
Flag	Result a b c d 377 (8)→000 0 0 1 1 (NOTE-2) Other than above 1 0 0 0 a: Non-carry 7354 b: Error 7355 c: Carry 7356 d: Zero 7357	

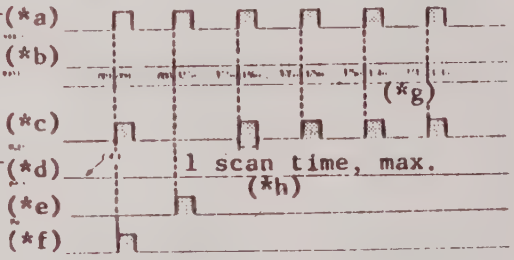
NOTE-1:].734 ~].737 is the special area.

NOTE-2: The contents of D are represented by a binary number, which can assume a number of 000 ~ 255 in decimal notation or 000 ~ 377 ($_8$) in the octal notation.

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

F-64 DEC — Decrement counter (Decrement)

Symbol		
Function	The contents of register D (binary data) are decremented.	(Explanation) When the input condition 0000 changes from OFF to ON, it decrements the data.
Operation	<D>-1→D	
Range of D applicable	].000~].733 (NOTE-2) b.000~b.377 9000~9377	
Operational condition	At the rising edge of signal (OFF to ON).	



(*a): Input
(*b): Register value
(*c): Non-carry flag
(*d): Error flag
(*e): Carry flag
(*f): Zero flag
(*g): 377 (8) indicates an octal number 377.
(*h): Valid until a flag affecting instruction is met in the program.

Contents of D	Result (binary code)				
Flag	Result	a	b	d	d
	001(8)→000 (8)	1	0	0	1
	000(8)→377 (8)	0	0	1	0
	Other than above	1	0	0	0

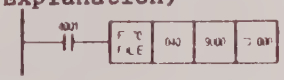
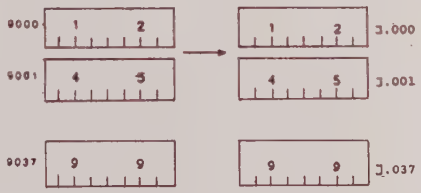
a: Non-carry 7354
b: Error 7355
c: Carry 7356
d: Zero 7355

NOTE-1:].734 ~].737 is the special area.

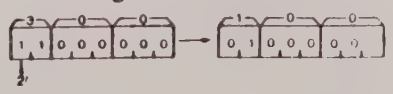
NOTE-2: The contents of D are represented by a binary number, which can assume a number of 000 ~ 255 in decimal notation or 000 ~ 377 ($_8$) in octal notation.

NOTE-3: State of the flag will be kept valid until the instruction that affects a next flag in that scan cycle is met.

F-70 FILE Transfer n-byte in batch (FILE)

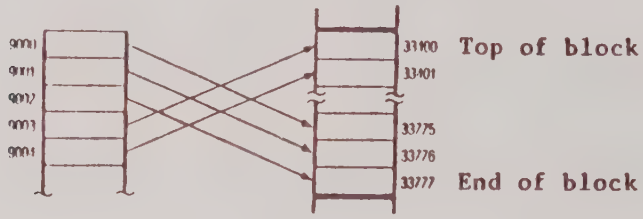
Symbol		
Function	A n-byte octal data from register S to the register S+n-1 are transferred in batch to the n-byte area beginning from reg D to D+n-1	(Explanation)  Instr. STR 4001 F-70 040 9000].000 When the input condition 4001 changes from OFF to ON, 040(8) bytes data (32 bytes in decimal) in register 9000 through 9037 are transferred in batch to the 32 bytes area of].000 through].037. The contents of registers, 9000~9037, remain unaffected. Before operation / After operation 
Operation	S, S+n-1 → D, D+n-1	
Range of "n" applicable	000~200 (8) (To be 128 bytes, if 000.)	
Range of S applicable	].000~].733 (Note-1) b.000~b.377 9000~9377 30000~33777	
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377 30000~33777	
Operational condition	At the rising edge of signal (OFF to ON).	
Contents of S- S+n -1	Unchanged	
Contents of D Contents of D+1 Contents of D+n-1	Contents of the register S Contents of the register S+1 Contents of the register S+n-1	
Flag	Unchanged	

NOTE-1:].734 ~].737 is the special area.

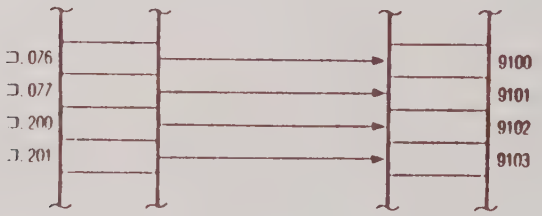
NOTE-2: Use an octal number within a range of 000 to 200 to represent the size of transfer bytes "n".
Though it is possible to use 300 ($_8$) in terms of programming, it will be taken for 100 ($_8$) as the bit in 2^7 is disregarded.
"n" designated 

NOTE-3: If "n", S, and D were programmed in such a way that extend over another block in the data memory block, the data is entered beginning at the top address of the same block.

Programming in the following manner will result in:

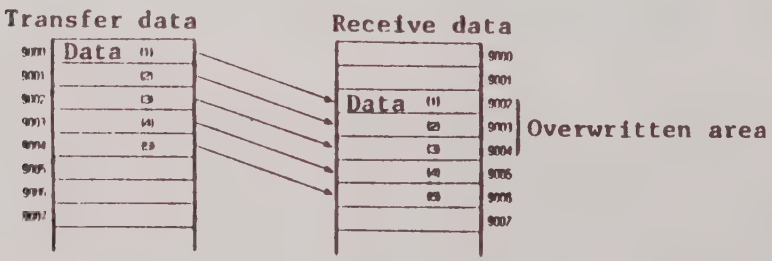


NOTE-4: If the].xxx area is used for S or D, transfer will take place by padding the unused relay numbers. It results in the following when programmed in this way, as].100 ~].177 are unused.

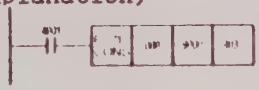
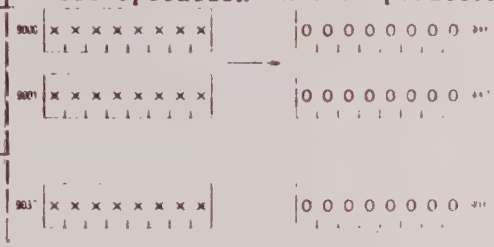


NOTE-5: File register area, 3000 ~ 33777, can be used when the file register capacity has been designed in the system memory (#205).

NOTE-6: It is possible to program "n", S, and D so that the source may overwrite the destination.

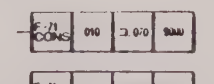


F-71 CONS Transfer octal constant in batch (Constant)

Symbol		
Function	An octal constant "n" transferred in batch from the register D1 to the register D2.	(Explanation)  Instr. STR 4001 F-71 000 9000 9037 When the input condition 4001 changes from OFF to ON, the octal constant 000 is transferred in batch from registers 9000 through 9037.
Operation	n→DD, ,D2 n=000~377(8)	
Range of "n" applicable	000~377(8)	
Range of D1 applicable	].000~].733 (Note-1) b.000~b.377 9000~9377 30000~33777	 Before operation After operation
Range of D2 applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377 30000~33777	
Operational condition	At the rising edge of signal (OFF to ON).	
After the operation: Contents of D1 / Contents of D1+1 / Contents of D2+1 / Contents of D2	Constant n	
Flag	Unchanged	

NOTE-1:].734 ~].737 is the special area.

NOTE-2: No operation takes place if D_1 or D_2 was programmed in such way as to extend over another block of the data memory.



The end address of the block that contains].070 is].737.

The end address of the block that contains 31000 is31377.

NOTE-3: If the address designation of D_1 is smaller than D_2 the instruction will not execute.

D1(9200)>D2(9000)

NOTE-4: If the].xxx area is used for D_1 or D_2, transfer will take place by padding the unused relay numbers

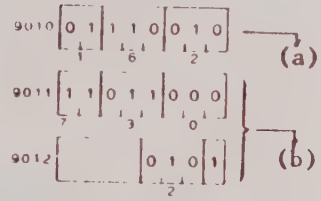
Programming in this way makes the octal constant 020 transfer to].000 ~].077,].200 ~].277, and].400 ~].410.

NOTE-5: File register area, 30000 ~ 33777, can be used when the file register capacity has been designed in the system memory (#205).

Symbol	
Function	The contents is n-byte register group whose address is obtained by modifying the top address of the data memory block in which S is contained (reference address) by the contents of S is transferred to n-byte file register group whose leading address is obtained by modifying 30000 by the contents of D and D+1 (file data pointer).
Operation	X+<S>,...X+<S>+n-1 →30000+D,D+1>,... 30000+<D,D+1>+n-1 X: Top address of the data memory block in which S is contained (reference address) <S>:Data pointer <D,D+1>:File data pointer
Range of "n" applicable	000~200(8) (128 bytes are established, if 000.)
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377
Operational condition	At a rising edge of input signal (OFF to ON)
Other than file register	Unchanged
30000+<D,D+1>	Contents of register X+<S>
30000+<D,D+1>+1	Contents of register X+<S>+1
30000+<D,D+1>+n-1	Contents of register X+<S>+n-1
Flag	Unchanged

Instr.
STR 4000
F-72 015
9010
9011

When the input condition 4000 changes from OFF to ON, the contents of the 015(8) bytes register group whose leading address is determined by modifying 9000 (the top address of the data memory block in which 9010 is contained) by the contents of 9010 are transferred to the 015(8) bytes file register group whose leading address is determined by modifying 30000 by the contents of 9011 and 9012 (file data pointer).



Reference addr.

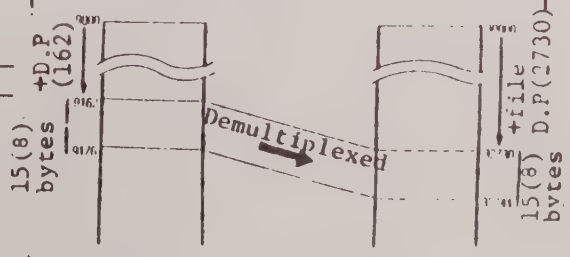
(a):
162(8) Has 9162 for leading address

9000+162
Data pointer
Reference address (top address of the block in which 9010 is contained)

(b):
2730(8) Has 32370 for the leading address

30000+2730
File data pointer

Reference addr.



NOTE-1:].734 ~].737 is the special area.

NOTE-2: The reference address on the register side is].000, b.000, 9000 which is the top address of the data memory block in which S is contained.

NOTE-3: F-72 may be used when the file register capacity has been set in the system memory (#205).

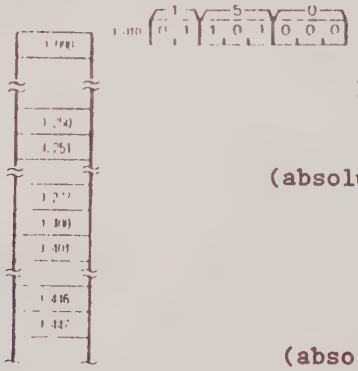
NOTE-4: Use an octal number of 000 ~ 200 ($_8$) for the transfer number "n". Although it is possible to program 217($_8$), it will be assumed to be 017($_8$) by disregarding the bit position of 2^7.

"n" designated

If "n" is programmed to 000($_8$), operation takes place assuming it to be 200($_8$).

NOTE-5: When the].xxx area is used for S, the leading address is determined by the absolute address having].000 for its regerence address and the range of transfer is also determined by the absolute address.

Program:



...Data pointer 150 ($_8$)

Reference address

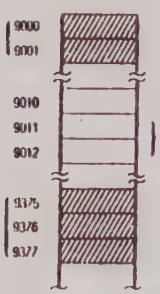
(absolute address 150 is].250)

(absolute address 247 is].447.)

NOTE-6: File data pointer is indicated by low order 12 bits of 2-byte area of D, D+1 (16 bits). High order 4 bits of D+1 are disregarded.

NOTE-7: If (reference address + data pointer + n-1) were to exceed the last address of the data memory block, the control returns to the top address of the same block.

(1) Register side example



Reference address

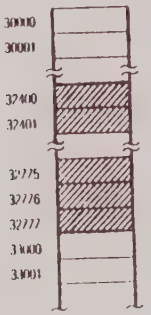
Data pointer

File data pointer

Contents of 9000+9010 (data pointer)

When the contents of 9010 (data pointer) is $375_{(8)}$, three bytes contents of 9375, 9376, and 9377 and two bytes conents of 9000 and 9001 are transferred to the file register.

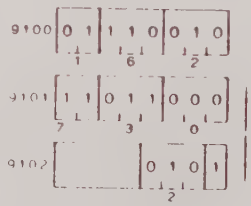
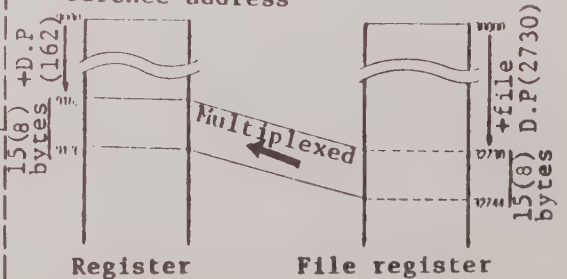
(2) File register side example



Top address of the block in which 32775 is contained.

The contents of 30000+9011 and 9012 (file data pointer).

When the contents of 9011 and 9012 are $2775_{(8)}$, register contents are transferred to three bytes of 32775, 32776, and 32777 and two bytes of 32400, 32401.
It does not extend to 33000 and 33001.

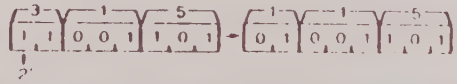
Symbol		
Function	The contents if n byte file reister group whose top address is obtained by modifying 30000 by the contents of S, S+1 (file data pointer) are transferred to the n-byte register group whose top address is determined by modifying the top address of the data memory block in which D is contained (reference address) by the contents of D (data pointer).	When the input condition 4000 changes from OFF to ON, the contents of the 015(8) bytes file register whose top address is determined by modifying 30000 by the contents of 9101 and 9102 (file data pointer) are transferred to the 015(8) bytes register group whose top address is determined by modifying 90000 (the top address of the data memory block in which 9100 is contained...reference address) by the contents of 9100 (data pointer).
Operation	30000+<S,S+1>,...1 30000+<S,S+1>+n-1 →X+<D>,...X+<D>+n-1 <S,S+1>..File data pointer X..Top adderss of the data memory in which D is contained (reference address) <D>..Data pointer	 (a) (b)
Range of "n" applicable	000~200(8) (128 bytes are established, if 000.)	Reference addr. (a): 162(8) Has 9162 for leading address 9000+162 ↳ Data pointer ↳ Reference address (top address of the block in which 9010 is contained)
Range of S applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377	
Range of D applicable	].000~].733 (NOTE-1) b.000~b.377 9000~9377	
Operational condition	At a rising edge of input signal (OFF to ON)	(b): 2730(8) Has 32730 for the leading address 30000+2730 ↳ File data pointer
After the operation - File register	Unchanged	Reference address 
After the operation - X+<D>	Contents of file register 30000+<S,S+1>	
After the operation - X+<D>+1	Contents of file register 30000+<S,S+1>+1	
After the operation - X+<D>+n-1	Contents of file register 30000+<S,S+1>+n-1	
After the operation - Flag	Unchanged	

NOTE-1:].734 ~].737 is the special area. Refer to 3-2[3].

NOTE-2: The reference address on the register side is].000, b.000, 9000 which is the top address of the data memory block in which D is contained.

NOTE-3: F-73 may be used when the file register capacity has been set in the system memory (#205).

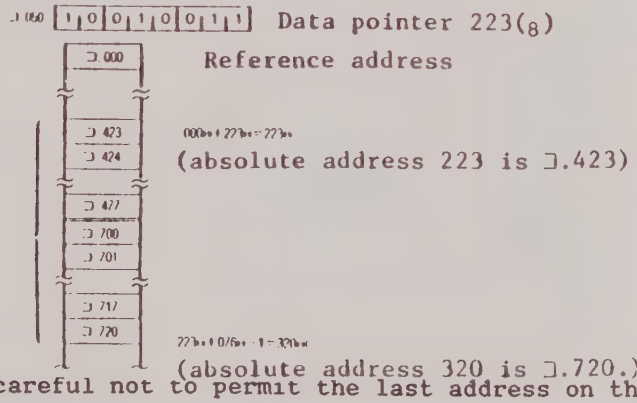
NOTE-4: Use an octal number of 000 ~ 200 ($_8$) for the transfer number "n". Although it is possible to program 315($_8$), it will be assumed to be 115($_8$) by disregarding the bit position of 2^7.

"n" designated 

If "n" is programmed to 000($_8$), operation takes place assuming it to be 200($_8$).

NOTE-5: When the].xxx area is used for D, the leading address is determined by the absolute address having].000 for its regerence address and the range of transfer is also determined by the absolute address.

Program:

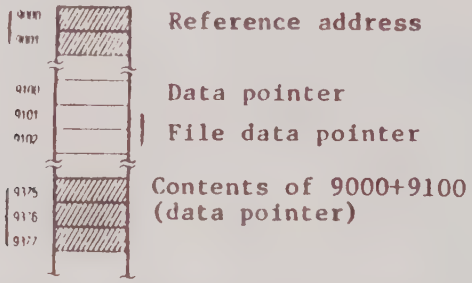


Be careful not to permit the last address on the register side not to exceed].733.

NOTE-6: File data pointer is indicated by low order 12 bits of 2-byte area of S, S+1 (16 bits).

NOTE-7: If (reference address + data pointer + n-1) were to exceed the last address of the data memory block, the control returns to the top address of the same block.

(1) Register side example



When the contents of 9100 is $375_{(8)}$, file register contents are transferred to three bytes of 9375, 9376 and 9377 and two bytes of 9000 and 9001.

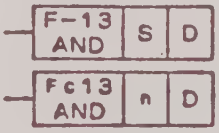
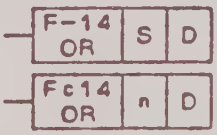
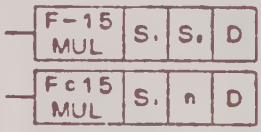
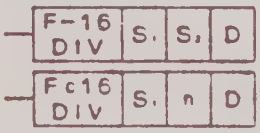
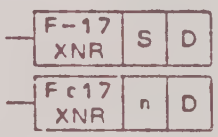
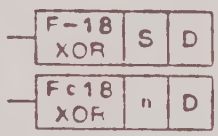
(2) File register side example

When the contents of 9101 and 9102 are $2375_{(8)}$, three bytes contents of 32375, 32376, and 32377 and two bytes contents of 32000 and 32001 are transferred to the register.
(It does not extend to 32400 and 32401.)

Application Instruction Summary

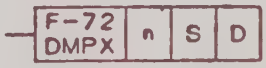
	Instruction and symbol	Number of words	Function
1	F-00 XFER | S | D	3	Transfer one byte data from register to register
2	F-01 BCD | n | D		Transfer BCD constant to a register n→D n=00∿99
3	F-02 XCH | D_1 | D_2		Exchange data between registers $D_1 \leftrightarrow D_2$
4	F-03 BCD-BIN | S | D		Convert from BCD into bianary (BIN) S→D
5	F-04 BIN-BCD | S | D		Convert from binary (BIN) to BCD S→D
6	F-05 DMPX | S | D		Demultiplex <S+1>→D+<S> D=□.000, b.000, 9000, 30000∿*
7	F-06 MPX | S | D		Multiplex S+<D>→D+1 S=□.000, b.000, 9000, 30000∿*
8	F-07 DCML | n | D		Transfer decimal constant n→D n=000∿255
9	F-08 OCT | n | D		Transfer octal constant n→D n=000∿377
10	F-09 INV | S | D		Invert and transfer data $\overline{S}$→D
11	F-10 ADD | S_1 | S_2 | D Fc10 ADD | S_1 | n | D	4	Add two register data S_1+S_2→D Add BCD constant to register data S_1+n→D n=00∿99
12	F-11 SUB | S_1 | S_2 | D Fc11 SUB | S_1 | n | D		Subtract register data by register data S_1-S_2→D Subtract register data by BCD constant S_1-n→D n=00∿99
13	F-12 CMP | S_1 | S_2 Fc12 CMP | S_1 | n	3	Compare two register data S_1<=>S_2 Compare register data with octal constant S_1<=>n n=000∿377

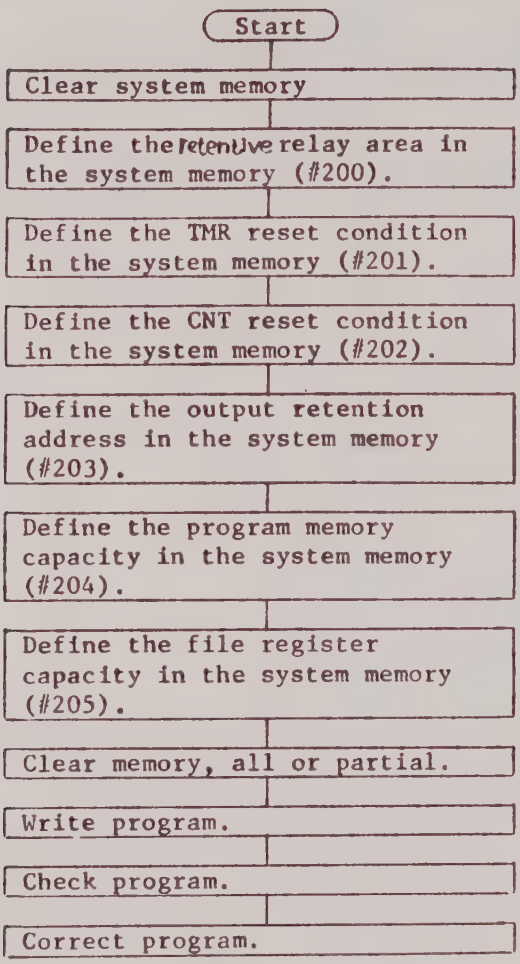
* Through file register setup

	Instruction and symbol	Number of words	Function
14		3	ANDs two register data $S \cap D \rightarrow D$ ANDs register data with octal constant $n \cap D \rightarrow D$ n=000∿377
15			ORs two register data $S \cup D \rightarrow D$ ORs register data with octal constant $n \cup D \rightarrow D$ n=000∿377
16		4	Multiply two register data $(S_1, S_1+1) \times (S_2, S_2+1) \rightarrow D, D+1, D+2, D+3$ Multiply register data by constant n=000∿999 $(S_1, S_1+1) \times n \rightarrow D, D+1, D+2, D+3$
17			Divide register data by register data $(S_1, S_1+1) \div S_2 \rightarrow D, D+1, D+2$ Divide register data by constant n=00∿99 $(S_1, S_1+1) \div n \rightarrow D, D+1, D+2$
18		3	Exclusive NORs two register data $\overline{S \oplus D} \rightarrow D$ Exclusive NORs register data with octal constant $\overline{n \oplus D} \rightarrow D$ n=000∿377
19			Exclusive ORs two register data $S \oplus D \quad D$ Exclusive ORs register data with octal constant $n \oplus D \quad D$ n=000∿377
20	F-30 MCS	1	Set master control
21	F-31 MCR		Reset master control
22	F-40 END		END
23	F-41 JCS		Set jump control
24	F-42 JCR		Reset jump control
25	F-43		Invert bit (negate operation section)
26	F-44		Positive going edge differenciation
27	F-45		Negative going edge differenciation

	Instruction and symbol	Number of words	Function
28	F-50 4→16 | S | D	3	Decode from 4 to 16 S→D, D+1
29	F-51 16→4 | S | D		Encode from 16 to 4(8) S, S+1→D
30	F-52 →7SEG | S | D		Decode into 7-segment S→D
31	F-53 BCD→BIN | S | D		BCD→BIN S, S+1→D, D+1 (4 digits of BCD into 16 bits of binary)
32	F-54 BIN→BCD | S | D		BIN→BCD S, S+1→D, D+1, D+2 (16 bits of binary into 6 digits of BCD)
33	F-55 SWAP | S | D		Swap high order 4 bits with low order 4 bits S→D
34	① ② ③ ④ F-60 F/B SFR | D	2	Shift register, bidirectional ① Shift direction ② Data input ③ Shift input ④ Reset input
35	① ② F-61 ASFR | D		Shift register, asynchronous ① Shift direction ② Shift input
36	① ② ③ F-62 U/D CNT | D		Set BCD up/down counter ① Direct up or down ② Count input ③ Reset input
37	F-63 INC | D		Increment counter
38	F-64 DEC | D		Decrement counter
39	F-70 FILE | n | S | D	4	Transfer "n" bytes in batch S,...S+n-1→D,...D+n-1 S, D=□.000₈, b.000₈, 9000₈, 30000₈* n=0₈200 (to be 128 bytes when 0)
40	F-71 CONS | n | D. | D.		Transfer octal constant in batch n→D_1, ..., D_2 D_1, D_2=□.000₈, b.000₈, 9000₈, 30000₈* n=000₈377

* Through file register setup

Instruction and symbol		Number of words	Function
41		4	Demultiplex file register X+<S>,...X+<S>+n-1→30000+<D,D+1>, ...30000+<D,D+1>+n-1 S, D=□.000∿, b.000∿, 9000∿ X: top address of a register group specified by "S" (□.000, b.000, 9000) n=000∿200 (to be 128 bytes when 0)
42			Multiplex file register 30000+<S,S+1>,...30000+<S,S+1>+n-1→ X+<D>,...X+<D>+n-1 S, D=□.000∿, b.000∿, 9000∿ X: top address of a register group specified by "S" (□.000, b.000, 9000) n=000∿200 (to be 128 bytes when 0)


Start
Clear system memory
Define the retentive relay area in the system memory (#200).
Define the TMR reset condition in the system memory (#201).
Define the CNT reset condition in the system memory (#202).
Define the output retention address in the system memory (#203).
Define the program memory capacity in the system memory (#204).
Define the file register capacity in the system memory (#205).
Clear memory, all or partial.
Write program.
Check program.
Correct program.

System memory		Initial state
Address (octal)	Contents after clear (octal)	
#200	300	Retentive relay area, 7000~7337
#201	000	TMR reset upon power failure
#202		CNT reset upon ON
#203		All outputs retained during stop
#204	001	Program memory capacity at 3.5KW
#205	000	File register not used

Fig. 8, Programming Sequence - PC 0185

All of the defined conditions (system addresses 200 to 205) will be returned to their initial states when the system memory is cleared. It is not necessary to define conditions that will be used in their initial states.

Clear the system memory after the program memory and the file register sizes are set.

Addressing Procedure

1. Program Memory Address Setup

Follow these instructions to specify the addresses to read from and write to, insert or delete the contents of, or search.

1. Push the ADRS key.
2. Push the 0 key until the address on the display reads all 0's.
3. Enter the desired address on the programmer using keys 0 thru 7 . (The address will be in octal)

See the following table for the relationship of the memory size to the programming address.

Memory capacity	Address range
2.5KW	00000~04777
3.5KW	00000~06777

Fig. 9, Memory Capacity/Address Range

After each key entry, the address on the display shifts one digit to the left. When the address exceeds the address range the high order digits turn to 0's, and the last key entry becomes the new address on the display. The example shown on the next page assumes a memory size of 2.5KW (5KB).

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00267 STR 0012	
[ADRS]		No change takes place in the display contents.
[0][0]	00000 STR 0012	Address display is cleared.
[5]	00005 STR 0012	
[1]	00051 STR 0012	
[2]	00512 STR 0012	
[3]	00003 STR 0012	Address range (00000~04777) is exceeded.
[MNTR]	00003 AND 0246	The contents of the address 00003.
[CLR]	00000	Display cleared.
[ADRS][2][6][7]	00267	Address entered.
[MNTR]	00267 STR 0012	The contents of the address 00267.
[3]	00267 STR 0123	Data memory number changed.

Fig. 10, Exceeding Address Range

When a numeric key is pressed instead of the ADRS key (while the contents are on the display), it will be seen as a data memory number; and displayed in the data display section.

Address setup is also available in the CHANGE and MONITOR modes.

2.System Memory Address Setup

Any system address within the range of].000 to].377 can be read from or written to in the following manner:

1. Push the CLR key, to clear the display.
2. Push the SYS MEM key. The address display shows 000.
3. Push the numeric keys (0 to 7) to specify the address.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR	00000	Display is cleared.
SYS MEM	#000	
2 0 0	#200	Address #200 is set.

Fig. 11, Address Setup

Program Mode

Prior to programming, the PC must be set in the PROGRAM mode by pressing the keys as shown below.

Fig. 12, Entering the PROGRAM Mode

Memory Clear

1. Memory Clear - except System Memory

The entire program area is padded with NOP instructions and an END command is written into the last address. The entire data memory area is reset to all 0's.

Configuration	After operation	2.5KW	3.5KW
Basic configuration	NOP	00000 ∫ 04776	00000 ∫ 06776
	END	04777	06777
	0	Data memory File register (30000 ∫ 33777)	Data memory

Fig. 13, Memory Clear Parameters

Operational Procedure:

Key operation	Display	Note
[*] [*] [PROG MODE] [0 SET]	76543210765432 10 02345 STR 0000	This operation is required even in the PROGRAM mode. There would be no change in the display.
[CLR]	00000	
[0 RST]	00000 MEM CL	
[ENT]	00000 MEM CL OK	

Fig. 14, Memory Clear Procedure

2. Partial Memory Clear

A. File Register Clear
The entire file register area can be cleared. The program and data memory areas remain unchanged.

After the clear instruction, all the following file registers are cleared.

Key operation	Display	Note
[*] [*] [PROG MODE] [0 SET] [CLR] [0 RST]	76543210765432 10 00000 MEM CL	
[0]	FILE MEM CL	
[ENT]	FILE MEM CL OK	

Fig. 15, Partial Memory Clear

3. System Memory Clear

The following procedure must be used to set the system memory to its initial state.

Key operation	Display	Note
[ ] [ ] [PROG MODE] [SET] [CLR] [RST]	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 OOOOO MEM CL	
[SYS MEM]	SYS MEM CL	
[ENT]	SYS MEM CL OK	

Fig. 16, System Memory Clear

System Memory Read and Write

1. Reading System Memory

1. Push the CLR key, to clear the display.
2. Push the SYS MEM key.
3. Use keys 0 thru 7 to enter the address.
4. Push the MNTR key to read the contents of the specified address.
5. Press the CONV key to change the displayed value from hex to octal to decimal to binary, and back to hex again.
6. Push the STEP (+) key to increment the address.
7. Push the STEP (-) key to decrement the address.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0.	
[CLR]	00000	Display is cleared.
[SYS MEM]	#000	
[2] [0] [0]	#200	System memory address is set.
[MNTR]	#200 HEX C0	The address is displayed in hexadecimal.
[CONV]	#200 OCT 300	Display changes to octal
[2] [6] [0] [ENT]	#200 OCT 260	An octal number of 260 is written in the address #200.
[STEP (+)]	#201 OCT 000	Address is incremented.
[0] [0] [1] [ENT]	#201 OCT 001	An octal number of 001 is written in the address #201.

Fig. 17, Reading System Memory

2. Writing to System Memory

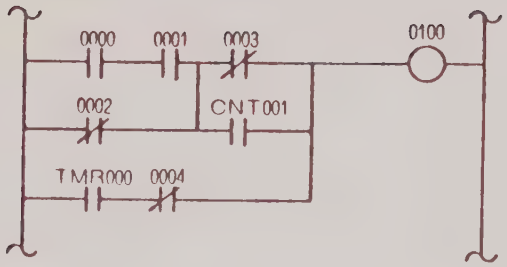
1. Use steps 1 thru 6 above to enter the starting memory address.
2. Use the 0 thru 7 keys to enter the data.
3. Push the ENT key to store the data in memory.

For the PC 0185:

Relay type	Relay address	One-byte address	Absolute address
I/O relay	0000 ∿ 0777	□000 ∿ □077	000 ∿ 077
Link relay	2000 ∿ 2777	□200 ∿ □277	100 ∿ 177
Auxiliary relay	4000 ∿ 4777	□400 ∿ □477	200 ∿ 277
Retentive relay	7000 ∿ 7337	□700 ∿ □733	300 ∿ 333

Note 1: The programmer can not write to system memory 100.

Example

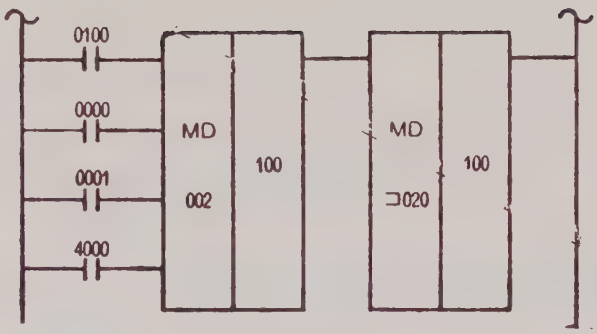


Instruction		
STR		0000
AND		0001
OR	NOT	0002
STR	NOT	0003
OR	CNT	001
AND	STR	
STR	TMR	000
AND	NOT	0004
OR	STR	
OUT		0100

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR 1 2 3	00123	Starting address or
CLR SRCH MNTR	00127 NOP	Starting from first unused address
STR ENT	00127 STR 0000	Enter STR0000. (Not necessary to enter 0000)
STEP (+) AND 1 ENT	00130 AND 0001	
STEP (+) OR NOT 2 ENT	00131 OR NOT 0002	NOT must follow OR.
STEP (+) STR NOT 3 ENT	00132 STR NOT 0003	
STEP (+) OR CNT 1 ENT	00133 OR C001	C of C001 represents CNT and 001 the CNT number.
STEP (+) AND STR ENT	00134 AND STR	Data section is not displayed.
STEP (+) STR TMR ENT	00135 STR T000	T of T000 represents TMR and 000 the TMR number.
STEP (+) AND NOT 4 ENT	00136 AND NOT 0004	
STEP (+) OR STR ENT	00137 OR STR	
STEP (+) OUT 1 0 0 ENT	00140 OUT 0100	

Fig. 20, Programming Example

Example



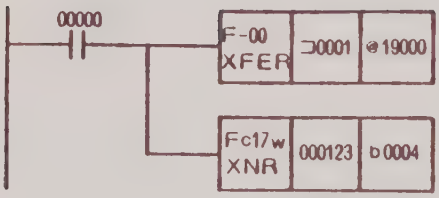
Instruction
STR 0100
STR 0000
STR 0001
STR 4000
MD 002
100
MD ⊐020
100

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR SRCH MNTR	00141 NOP	Searches the address where no program is written.
STR 1 0 0 ENT	00141 STR 0100	
STEP (+) STR ENT	00142 STR 0000	
STEP (+) STR 1 ENT	00143 STR 0001	
STEP (+) STR 4 0 0 0 ENT	00144 STR 4000	
STEP (+) MD	00145 MD (F-20) 000	000 is displayed in the display section.
2 ENT	00145 MD (F-20) 002	When the MD number has been entered, 000 is stored automatically in the next address as the preset value.
STEP (+)	00146 000	Reads the preset value stored in the preceding step.
1 0 0 ENT	00146 100	Stores the preset value 100.
STEP (+) MD CODE CONST	00147 MD (F-20) ⊐000	When the CODE CONST key is pressed immediately after MD, ⊐000 is displayed in the data display section.
2 0 ENT	00147 MD (F-20) ⊐020	⊐020 (0200∿0217) is specified for the MD output area.
STEP (+) 1 0 0 ENT	00150 100	

Fig. 22, Programming Example

Press the CODE/CONST key to determine from which area the MD preset should be output to.].000 for].000 to].177

Example



Instruction
STR 00000
F-00
□0001
@19000
Fc17w
000123
b0004

Fig. 23, Programming Example

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR SRCH INSTR	00164 NOP	
STR ENT	00164 STR 00000	
STEP (+) FUN ENT	00165 F-00 XFER	
STEP (+) 1 ENT	00166 □0001	
STEP (+) CONV	00167 @□0000	@ of @□0000 represents an indirect address.
CODE CONST CODE CONST CODE CONST ENT	00167 @19000	
STEP (+) FUN 1 7 CODE CONST CODE CONST	00170 F-17w XNR	w of F-17w represents the word processing instruction
CODE CONST ENT	00170 Fc17w XNR	
STEP (+) 1 2 3 ENT	00171 000123	
STEP (+) CODE CONST 4 ENT	00172 b0004	

Fig. 24, Programming Example

2. Checking the Program

There are two ways to check the program contents. One is to read the program after specifying a starting address. The other is to use the search instruction and data memory.

Checking by specifying a starting address:

1. Push the CLR key.
2. Enter the memory address to be read.
3. Push the MNTR key. The program contents at the specified address are displayed.
4. To continue reading, press the STEP (+) key to increment or press the STEP (-) key to decrement the address.

Address	Instruction	
00000	STR	0000
00001	AND	0001
00002	AND NOT	0002
00003	AND	0003
00004	OUT	2000
00005	STR	0000
00006	TMR	000
00007		0010
00010	STR TMR	000
00011	OUT	2001
00012	STR	0010
00013	OR	0011
⋮		
06775	MD	177
06776		123
06777	F-40	

Fig. 25, Program Check

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR 1 0 MNTR	00010 STR T000	Reads the contents of address 00010.
STEP (+)	00011 OUT 2001	
STEP (+)	00012 STR 0010	
CLR MNTR	00000 STR 0000	
STEP (+)	00001 AND 0001	
STEP (-)	00000 STR 0000	When the STEP (-) key is depressed with the address 00000 on display during continuous reading, it displays the highest memory address.
STEP (-)	06777 F-40 END	
STEP (-)	06776 123	
STEP (+)	06777 F-40 END	When the STEP (+) key is depressed with the highest address on display, it jumps to the address 00000.
STEP (+)	00000 STR 0000	
STEP (+)	00001 AND 0001	
STEP (+)	00002 AND NOT 0002	

Fig. 26, Program Check

Program Check

1. Checking Parity

The PC checks the parity down to the F-40 (END) instruction during each scan. The result is stored in the special relay 7370. An entire program can be checked for a parity error in the PROGRAM mode.

Use this procedure to check for parity errors:
1. Enter the PROGRAM mode.
2. Push the CLR key.
3. Push the * key twice.
4. Push the CLR key again. This starts the check.

Parity check	Display	Note
Normal	00000 CHECKED	
Error	00056 PARITY ERROR	A parity error occurred in the address 00056.

Fig. 27, Parity Check Display

5. When the CONT key is pressed while an error is indicated, the PC performs another check beginning at the address of the last error. If no parity error is found the display will read CHECKED.

To return to normal operation after a parity check:

1. If you are just beginning to program, clear the entire memory area.
2. Press the MNTR key and read the contents of the address where the error exists. Enter the correct instruction.
3. Press the CONT key to check the addresses of other errors. If there are too many errors, reload the program from the tape.

If a parity error still exists after following these procedures check special relay 7372 (battery failure). If the battery is OK, replace the memory of the controller.

2. Checking the Program

Before a trial run of the program, and after program revisions you should always check for errors. It is essential to correct any errors before the program is run.

A. Contents of the program check
   1. Stack operating status of instructions
   2. MCS/MCR operating status
   3. JCS/JCR operating status
   4. Double use of OUT instructions
   5. Double use of TMR, CNT, and MD instructions
   6. Omission of END instruction

B. Program check procedure:
   1. Enter the PROGRAM mode.
   2. Push the CLR key.
   3. Push the * key twice.
   4. Push the SRCH key to begin the search.
   5. When the CONT key is pressed while the current error is displayed, the PC will search for the next error.

C. Example:

	Address	Instruction		
	00000	STR		0000
	00001	STR	NOT	0001
*	00002	OUT		4000
	00003	STR		4001
	00004	OR		0002
	00005	F-41		
	00006	STR		0002
	00007	AND		0003
*	00010	F-41		
	00011	STR		0004
	00012	OUT		4002
*	00013	F-31		
	00014	STR		0005
	00015	AND	NOT	4003
	00016	OUT		0100
	00017	F-42		

* = errors in the program

Fig. 28, Program Being Checked

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00000	
SRCH	00002 STACK OVER	Overuse of STR(NOT) instruction.
CONT	00010 JCS ERROR	Double use of the F-41 (JCS) instruction.
CONT	00013 MCR ERROR	F-31(MCS) is used without preceding F-30(MCS).
CONT	00013 CHECKED MCR ERROR	Indicates existence of no program error after the address 00014.

Fig. 29, Program check Display

D. Program Message Error List

Message	Address	Significance	Action
CHECK OK	00000	No syntax error found in the program.	
STACK OVER	Address where stack over-flow is met.	Use of too many STR (NOT) instructions.	Delete STR(NOT) or insert AND(OR)STR.
STACK UNDER	Address where stack under-flow is met.	Short of STR(NOT) or too many use of AND (OR)STR.	Insert STR(NOT) or delete AND(OR)STR.
STACK EXIST	Address where F-40(END) is written.	Data are remaining in the stack even after F-40(END).	Insert or delete instructions.
MCR ERROR	Address where an MCR error is found.	F-31(MCR) is used without preceding F-30(MCS).	Delete F-31(MCR) or insert F-30(MCS).
MCS EXIST	Address where F-40(END) is written.	F-30(MCS) has not been reset before F-40(END).	Insert F-31(MCR).
JCS ERROR	Address where F-41(JCS) doubled.	Another F-41(JCS) is used within an effective range of F-41(JCS). Nesting is not permitted for F-41(JCS)	Delete F-41(JCS).
JCR ERROR	Address where a JCR error is found.	F-42(JCR) is used without preceding F-41(JCS).	Delete F-42(JCR) or insert F-41(JCS).
JCS EXIST	Address where F-40(END) is written.	F-41(JCS) has not been reset before F-40(END).	Insert F-42(JCR).
DOUBLE OUT	Address where the same output inst-ruction is found.	Double use of the same relay for output instruction (OUT).	Correct the relay number of the output instruction.
DOUBLE NUMBER	Address where the data memory is used double.	Double use of TMR, CNT, MD number.	Correct the TMR, CNT, MD number.
NO END ERROR	00000(W16/51) Last address (W100)	F-40(END) does not exist in the program.	Write F-40(END).

Fig. 30, Error Messages

Program Search

The search function finds the address where a particular instruction resides. Any instruction and data memory location can be searched.

0000 ∿ 07777	I/O relay
2000 ∿ 2777	Link relay
4000 ∿ 4777	Auxiliary relay
7000 ∿ 7377	Retentive relay (7340∿7377 are for special use)
000 ∿ 177	TMR, CNT, MD
□000 ∿ □737	Byte address, 0000∿7377 can be handled as register.
b000 ∿ b377	TMR, CNT, MD current byte address can be handled as register.
9000 ∿ 9377	Register
30000 ∿ 33777	File register (30000∿33777) 2K byte auxiliary register available when program memory area is reconfigured from 3.5K to 2.3K words.

Fig. 31, Memory Addresses

Search Direction and Key Operation

Incrementing Addresses: SRCH MNTR
Decrementing Addresses: SRCH STEP(-) MNTR

Repeat this procedure to continue searching. This feature works in all three modes.

1. Instruction Search

1. Push the CLR key.
2. Enter the starting search address.
3. Enter the instruction to be searched for.
4. Push the SRCH key.
5. Push the key that directs the search direction. (Press the MNTR key to begin searching from the starting address to the last address. Push the STEP (-) key and then MNTR to search from the starting address backwards to 0000.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
[CLR]	00000	Display is cleared.
[1][0][0][0]	01000	
[OUT][1][2][3]	01000 OUT 0123	
[SRCH][MNTR]	01567 OUT 0123	
[SRCH][MNTR]	01605 OUT 0123	
[SRCH][STEP (-)][MNTR]	01567 OUT 0123	
[SRCH][STEP (-)][MNTR]	01567 NOT. FOUND	Indicates that the given instruction does not exist in the address decrementing direction.

Fig. 32, Searching for Instructions

2. Data Memory Search

It is possible to find the data memory address without specifying the instruction. For example, you can search for the address that has internal relay 4000.

1. Push the CLR key.
2. Push the CODE/CONST key and enter the type of memory to be searched.
3. Enter the data memory to be searched for.
4. Push the SRCH key.
5. Push the key that directs the search direction. (Press the MNTR key to begin searching from the starting address to the last address. Push the STEP (-) key and then MNTR to search from the starting address backwards to 0000.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR	00000	Display is cleared.
CODE CONST	00000 0C00	Relay 0000 is set.
CODE CONST 1 2 3	00000 123	TMR, CNT, MD numbers are set.
CODE CONST CODE CONST CODE CONST 2 5 6	00000 9256	Register number is set.
CODE CONST CODE CONST 2 0 0 0	00000 2000	Relay number is set
SRCH MNTR	01013 OR 2000	Searches the given data memory.
SRCH MNTR	01254 AND NOT 2000	Search is continued.

Fig. 33, Data Search

Each time the CODE/CONST key is pressed after pressing the CLR key, the display's data is changed:

0000 (relay) to 000 (TMR,CNT,MD) to].000 to
b.000 to 9000 to 30000 to 0000 (relay)

Program Correction

Program revision is inevitable because of changes to the ladder diagram, key entry errors, and errors that are found during the program check.

1. Get the address and instruction to be corrected on the display by using the read and search functions.
2. Correct the instruction or number.
3. Enter the correction.
4. Check the instruction in front of, and behind, the correction.

1. Changing Instruction

Key operation	Display
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
After search	01023 OR NOT 2013
[OR]	01023 OR 0000
[2] [0] [1] [3]	01023 OR 2013
[ENT]	01023 OR 2013
After search	00056 TMR 012
[CNT] [1] [3] [ENT]	00056 CNT 013
[STEP (+)]	00057 0000
[1] [5] [ENT]	00057 0015

Fig. 34, Changing Instructions

2. Changing Data Memory, preset value, etc.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 01023 OR NOT 2013	
After search		
[2] [0] [1] [4] [ENT]	01023 OR NOT 2014	2013 is corrected to 2014.
After search	00056 TMR 012	
[STEP (+)]	00057 0050	The preset value is read.
[0] [1] [0] [ENT]	00057 0010	0050 is corrected to 0010. Pay attention for the number in the data section.

Fig. 35, Changing Data Memory

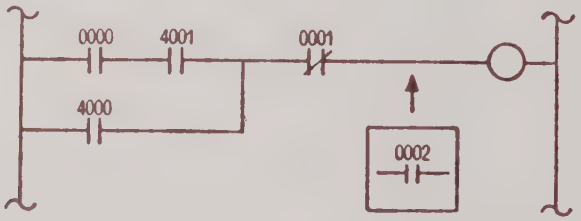
After the instruction or numeric value is changed, be sure to check the display. It should show the corrected value.

3. Inserting an Instruction

This procedure will insert a new instruction into program memory:

1. Bring the memory address or its contents onto the display by using the addressing, searching, or reading functions.
2. Enter a new instruction on the display.
3. Push the INS key.

Example: (Inserting an ADD condition)



Address	Instruction	
00100	STR	0000
00101	AND	4001
00102	OR	4000
00103	AND NOT	0001
00104	OUT	4000

→

Address	Instruction	
00100	STR	0000
00101	AND	4001
00102	OR	4000
00103	AND NOT	0001
00104	AND	0002
00105	OUT	4000

Key operation	Display
CLR OUT 4 0 0 0 SRCH MNTR	00104 OUT 4000
AND 2	00104 AND 0002
INS	00104 AND 0002
STEP (+)	00105 OUT 4000

Fig. 36, Inserting Instructions

Each press of the INS key causes all instructions after this step to increment one step.

Insertion will not occur if it would go beyond the programming capacity of the PC 0185.

Insertion is not possible for the 2nd, 3rd, or 4th byte of a 2, 3, or 4 byte instruction.

4. Deleting an Instruction

The following describes how to delete an instruction from the program memory:

1. Find the instruction to be deleted by using the addressing, searching, or reading functions.
2. Push the DEL key just once.

Example: (delete contact at step 3)

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR OUT 4 0 0 0 SRCH	00105 OUT 4000	
MNTR		
STEP (-)	00104 AND 0002	instruction to be deleted.
DEL	00104 OUT 4000	As the instruction of AND0002 is deleted, all instructions after OUT4000 are moved backward.

Fig. 37, Deleting an Instruction

Each press of DEL will cause all of the instructions after this step to decrement (move towards 0000).

Insertion is not possible for the 2nd, 3rd, or 4th byte of a 2, 3, or 4 byte instruction.

Recording and Verifying

It is possible to record the program memory and data memory contents onto cassette tape and verify it in any of the three operating modes.

Recording

Connect the tape recorder MIC jack with the cassette with the interfacing cable. The cable is included with the accessories.

1. Push the CLR key.
2. Push the REC key.
3. Push the 0 key to record the data memory or a 1 to record the program memory.
4. Start the tape recorder.
5. Push the 8/SET key to begin recording.
6. Push the CLR key to stop recording.
7. Stop the tape recorder.
8. Push the CLR key.
9. Verify the recording using the procedure in the next section.

CLR REC 0 8/SET	Data memory (including the file register) (NOTE-1)
CLR REC 1 8/SET	Program memory System memory

Fig. 38, Recording a Program

The following procedure describes how to record the program memory:

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 00000	
CLR		
REC	RECORD DATA (0), PROGRAMME (1) ?	Push the REC key to record the program memory.
1	RECORD PUSH SET KEY	Start cassettes in record mode.
SET	IRG REC START	
	MARK IN REC 1-	
	00000 IN REC 1-	The address of the memory being recorded is shown in the address section.
	01234 IN REC 2-	In second recording
	00057 IN REC 3-	In third recording
CLR	00162 TAPE STOP 3-	Recording is stopped in a middle of the third operation.
CLR	00000	Display returned to its initial state.

Fig. 39, Recording Program Memory

The entire program memory must be recorded. Be sure to keep a written record of the contents of each tape.

Verifying a Program

The contents of the tape can be verified against the contents of the PC memory by following this procedure:

1. Place the tape to be verified in the tape recorder.
2. Connect the recorder's earphone jack with the cassette using the interface cable.
3. Push the CLR key.
4. Push the 0 key to record the data memory or a 1 to record the program memory.
5. Push the 8/SET key to begin recording.
6. Start the tape recorder and playback the tape.
6. Push the CLR key to stop recording.
7. Stop the tape recorder.
8. Push the CLR key.

If no change takes place in the display the tape recorder volume level may be too low. Adjust the volume of the tape recorder until the ON/OFF display reads ON; and begin verification again.

Procedure for verifying a program:

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00000	
CLR		
VERI	VERIFY DATA(0), PROGRAMME(1) ?	Push the 1 key to verify the program memory.
1	VERIFY PUSH SET KEY	
SET	VER START ■	Start the tape recorder playback mode.
	MARK IN VERIFY 1- ■	
	00000 IN VERIFY 1- ■	The address section increments.
	00000 IN VERIFY 1-1 (OK) ■	First verification successful.
	01023 IN VERIFY 3-2 (OK) ■	When a verify error is met, the display stops here and waits for a next verification.
	00000 IN VERIFY 4-3 (OK) ■	It is now under the fourth verification and indicates the the third verification has been successful.
	00013 TAPE STOP 5-3 (OK)	Verification stops.
CLR		
	00000	The display returns to its initial state.
CLR		

Fig. 40, Program Verification

Saving a Program

General Information

1. Time required to record a program:

Program Capacity	Time Needed for One Transfer
3.5 KW	3 Mins. : 20 Secs.

Record the program five (5) times to avoid accidental loss.

2. Cassette Tape Specification:

Type of tape	Nominal record/ playback time (4.76 cm/sec)
C90	90 minutes
C60	60 minutes
C46	46 minutes

3. Cassette Recorder Specifications:

Recording method	AC biased
Erasure method	AC biased
Wow and Flutter	.2% max
Output jack	Earphone jack, JIS C 6560, 3.50
Input jack	Microphone jack, JIS C 6560, 3.50
Tape speed error	+/- 2% max
Other	Must have ALC feature

4. Record/Playback Specifications

Transfer speed	600 bits per second
Playback input	1V RMS min
Input impedance	200 Ohms
Recording output	10mV RMS min
Output impedance	200 Ohms
Check	CRC

5. Cautions

Use the same model of tape recorder to record and play back the tape.

Verify the tape after recording.

The write-protect tab must be broken to prevent accidental erasure.

The tape speed should be constant. Avoid using battery powered recorders.

Loading a Program

Programs or data recorded on tape can be loaded into the programmer memory. Loading is permitted only in the PROGRAM mode.

1. Place the tape to be loaded into the recorder.
2. Connect the recorder's earphone jack with the cassette using the interface cable.
3. Push the CLR key.
4. Push the LOAD key.
5. Push the 0 key to load the data memory or the 1 to load the program memory.
6. Push the 8/SET key to start the process.
7. Start the tape and play back the recording.
8. Confirm that LOAD DONE appears on the display.
9. Stop the tape recorder.
10. Push the CLR key.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0	
CLR	00000	
LOAD	LOAD DATA(0), PROGRAMME(1) ?	
1	LOAD PUSH SET KEY	
8 SET	LOAD STRT	Start the tape recorder to play back. the ON/OFF indicator is flickering.
	MARK IN LOAD 1-	
	00056 IN LOAD 1-	
	00000 LOAD DONE 1-	
CLR	00000	

Fig. 41, Loading a Program

The entire data memory is cleared after program memory loading is complete.

Push the CLR key to stop loading prematurely.

Monitor Mode

In the MONITOR mode the PC can monitor program contents, relay ON/OFF states, and TMR and CNT current values. Press these keys to enter the MONITOR mode:

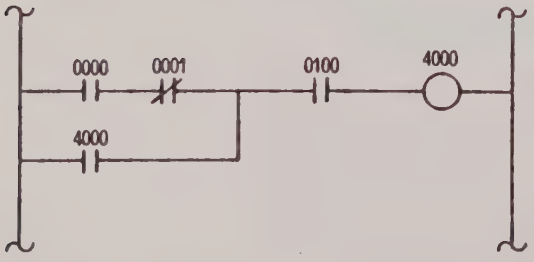
The MONITOR indicator should turn ON.

Program Monitor

1. Monitoring Relays
   There are three ways to read the program contents and monitor the ON/OFF state of relays:
   a. Specify the address
   b. search for the instruction
   c. search the data memory

   1. Specifying the Address
      1. Push the CLR key.
      2. Enter the memory address to be read.
      3. Push the MNTR key. The program contents will be displayed.
      4. To continue reading, press the STEP (+) key or the STEP (-) key.

   While the program is being monitored, the state of the coil and contact point is displayed and updated on each scan. The following example assumes that latched relays 0000, 0001, 0100, and 4000 are ON.



Address	Instruction	
00010	STR	0000
00011	AND NOT	0001
00012	OR	4000
00013	AND	0100
00014	OUT	4000

Fig. 42, Monitor Logic Example

Key operation	Display	Note
CLR 1 0	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00010	The address is specified.
MNTR	00010 STR 0000•	The contents of address 00010 are read.
STEP (+)	00011 AND NOT 0001□	□ is displayed to indicate the use of b-contact point.
STEP (+)	00012 OR 4000•	
STEP (+)	00013 AND 0100•	
STEP (+)	00014 OUT 4000•	

Fig. 43, Monitor Procedure

Even though coil 0001 is in the ON state, address 0001 will display the OFF indicator; which indicates circuit continuity.

ON/OFF indication will not be shown when an instruction other than STR, AND, OR, and OUT is given.

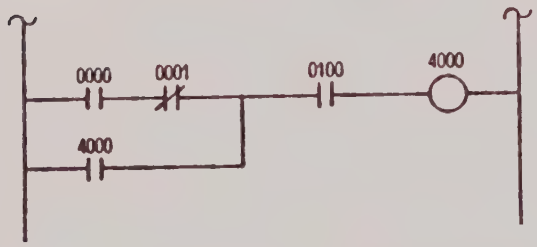
Operational flags (7354 to 7357) are indicated OFF on the display regardless of the actual operating condition.

2. Searching for the Instruction

Search direction and key operation:
Incrementing address: SRCH MNTR
Decrementing address: SRCH STEP (-) MNTR

Repeat this procedure to continue searching. This feature is available in all three modes.

Searching for a specific instruction:
1. Push the CLR key.
2. Enter the starting search address.
3. Enter the instruction to be searched for.
4. Push the SRCH key.
5. Push the key that directs the search direction. (Press the MNTR key to search from the starting address to the last address. Press the STEP (-) key and then the MNTR key to search from the starting address to address 0000.



Address	Instruction	
00010	STR	0000
00011	AND NOT	0001
00012	OR	4000
00013	AND	0100
00014	OUT	4000

Fig. 44, Monitor Logic Example

Key operation	Display	Note
CLR OUT 4 0 0 0	00000 OUT 4000	The instruction is entered.
SRCH MNTR	00014 OUT 4000	Searching is done in the address incrementing direction from the address 00000.
STEP (-)	00013 AND 0100	Read is done from the address decrementing direction
STEP (-)	00012 OR 4000	
STEP (-)	00011 AND NOT 0001	
STEP (-)	00010 STR 0000	

Fig. 45, Monitor Example

3. Searching the Data Memory

1. Push the CLR key.
2. Push the CODE/CONST key and enter the type of data memory to be searched for.
3. Enter the data memory number.
4. Push the SRCH key.
5. Push the key that directs the search direction. (Press the MNTR key to search from the starting address to the last address. Press the STEP (-) key and then the MNTR key to search from the starting address to address 0000.

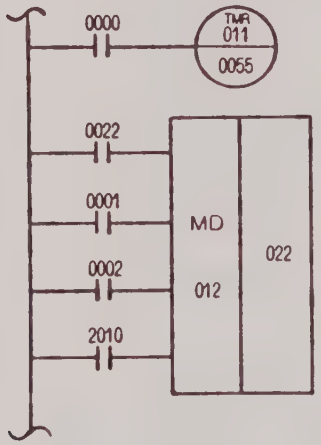
Key operation	Display	Note
[CLR] [CODE/CONST] [4] [0] [0] [0]	76543210765432 10 00000 4000	Searches the data memory 4000
[SRCH] [MNTR]	00012 OR 4000■	
[SRCH] [MNTR]	00014 OUT 4000■	Repeats searching the address which uses data memory 4000.
[SRCH] [STEP (-)] [MNTR]	00012 OR 4000■	Search is done in the address decrementing direction.

Fig. 46, Monitor Example

2. Monitoring TMR, CNT, and MD

TMR, CNT, and MD current values can be monitored at the same time the contents of the program are read.

Example:



Address	Instruction
00016	STR 0000
00017	TMR 011
00020	0055
00021	STR 0022
00022	STR 0001
00023	STR 0002
00024	STR 2010
00025	MD 012
00026	022

S_3(0022)·····OFF
S_2(0001)·····OFF
S_1(0002)·····ON

MD monitor display

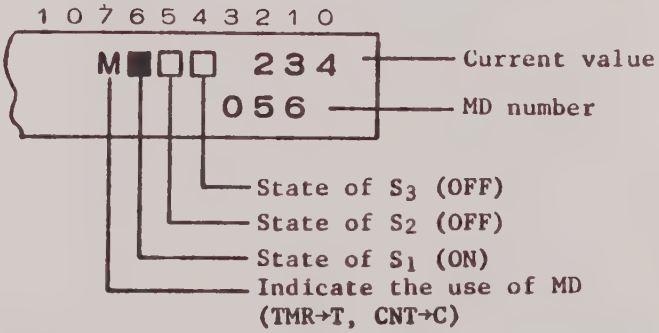


Fig. 47, Monitoring TMR, CNT, and MD

If the state of the MD output direction has not been set ON (2010 in this example) the current value may be 'T' (indicating TMR).

When a different state (T, C, M) is displayed for the current instruction, check the program for a double number entry.

Key operation	Display	Note
CLR 1 7 MNTR	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00017 T 0051 TMR 011	Indicates that the TMR011 current value is 51.
STEP (+)	00020 0055	
STEP (+)	00021 STR 0022▫	
STEP (+)	00022 STR 0001▫	
STEP (+)	00023 STR 0002▪	
STEP (+)	00024 STR 2010▪	
STEP (+)	00025 M▪▫▫ 022 MD (Note-1) 012	Indicates that S_2 and S_3 are OFF and S_1 ON. The current value is 22.
STEP (+)	00026 022	

Fig. 48, Monitor Procedure for TMR, CNT, and MD

3. Monitoring a Register

The current value of the register can be monitored at the same time as the program contents.

1. Push the CLR key.
2. Enter the memory address to be read.
3. Push the MNTR key. The program contents at the specified address are displayed.
4. To continue reading, press the STEP (+) key or the STEP (-) key.

Repeatedly pressing the CONV key will cause the current value to change from BCD to octal to decimal to binary; and back to BCD.

Key operation	Display	Note
CLR CODE CONST CODE CONST CODE CONST 1 0	76543210765432 10 00000 ⊐010	
SRCH PNTR	00201 HEX 93 ⊐010	The current value of the register ⊐010 is 93 in the hexadecimal notation.
CONV	00201 OCT 223 ⊐010	Changed to octal notation.
CONV	00201 DCML 147 ⊐010	Changed to decimal notation.
CONV	00201 ■□□■□□■■ ⊐010	Changed to bit pattern.
STEP (+)	00202 ■■■■□■■■ 9023	Read is done in the address incrementing direction. Indicates the bit pattern.
CONV	00202 HEX F7 9023	
STEP (-)	00201 HEX 93 ⊐010	Read is done in the address decrementing direction.

Fig. 49, Monitoring Registers

Data Memory Monitor

Relay ON/OFF state; TMR, CNT, MD current values; and current register values can be monitored from a data memory address.

1. Push the CE key. Do not push the CLR key.
2. Push the CODE/CONST key repeatedly to set the data memory area to be monitored. The display will cycle through the memory map locations.
3. Enter the data memory number.
4. Push the MNTR key to display the ON/OFF state or current value.

1. Monitoring Relays

Example: (monitoring data memory 0000 to 7377)

Key operation	Display	Note
Program monitor	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00013 STR 0023□	
CE	00013	The instruction display section is cleared, but the address section is not cleared.
CODE CONST	00013 0000	Data memory is set in the relay area.
4 3 5 6	00013 4356	Relay number is entered. It is not required for the relay 0000.
MNTR	00013 4356□	The relay 4356 is OFF.
STEP (+)	00013 4357■	The relay 4357 is ON.
STEP (+)	00013 4360■	
STEP (−)	00013 4357■	
CE	00013	
MNTR	00013 STR 0023□	

Fig. 50, Relay Monitoring

Since the address display section is not cleared when the CE key is first pushed, it is possible to return from the data memory monitor mode to the program monitor mode as shown in the previous example.

When the operational flag (7354 to 7357) is monitored, it will show as OFF regardless of its actual state.

2. Monitoring TMR, CNT, and MD

Example: (monitoring the value of data memory addresses 000 to 177)

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00002	
CE		
CODE CONST CODE CONST	00002 000	
1 2 3	00002 123	
MNTR	00002 T 1853 123	The current value which the TMR is used in the data memory 123 is 1853.
STEP (+)	00002 M□■□ 254 124	Only S_2 of MD is ON.
STEP (+)	00002 T 0000 125	
STEP (+)	00002 C 0000 126	CNT126 is counted up.

Fig. 51, Monitoring Data Memory

3. Monitoring Register Values

The following procedure monitors registers in data memory addresses].000 to].377, b.000 to b.377, 9000 to 9377, or 30000 to 33777. To change the data memory area being displayed push either the CE or CLR key.

Key operation	Display	Note
	76543210765432 10	
CE	00003	The instruction display section is cleared.
CODE CONST CODE CONST CODE CONST	00003 ⊐000	Data memory area is entered.
2 3	00003 ⊐023	Data memory number is entered.
MNTR	00003 HEX 15 ⊐023	The current value of ⊐023 is 15 in the hexadecimal notation.
STEP (+)	00003 HEX 5E ⊐024	Data memory address is monitored in the address incrementing direction.
CONV	00003 OCT 136 ⊐024	Changed from the hexadecimal notation to the octal notation.
CE	00003	The instruction display section is cleared.
CODE CONST CODE CONST CODE CONST CODE CONST CODE CONST	00003 9000	
2 5 6	00003 9256	
MNTR	00003 HEX 32 9256	
STEP (-) CONV CONV CONV	00003 □□■■□□■□ 9255	Displayed in the bit pattern mode in the data memory address decrementing address.

Fig. 52, Monitoring Registers

Change Mode

All the functions in the MONITOR mode are available in the CHANGE mode. The CHANGE mode functions are:

a. select latched relays,
b. TMR and CNT set/reset,
c. change current register values,
d. change TMR, CNT, and MD presets, and
e. change the value of a constant.

To enter the CHANGE mode.

The CHANGE indicator will be ON.

Setting and Resetting Latched Relays

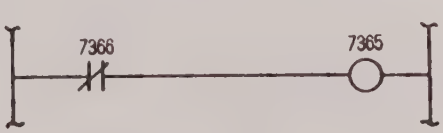
It is possible to set and reset latched relays 7000 thru 7337:

1. Monitor the latched relay as described on page 3-35.
2. Push the 8/SET key to set the latched relay; or push the 9/RST key to reset the latched relay.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00021 7010□	The latched relay 7010 is monitored.
	00021 7010▪	The latched relay 7010 is set.
	00021 7010□	The latched relay 7010 is reset.

Fig. 53, Setting/Resetting Latched Relays

Relays other than latched can be reset only if the preset value change coil (7365) is ON. 7365 is cleared after each scan; therefore it must be programmed with the Ever-ON contact, as shown:



Set or reset can only be executed in the operational period immediately after key entry. If the latched relay is used for an output instruction, it can be set or reset depending on the operational result.

Setting/Resetting TMR, CNT, MD

The following describes how to reset the TMR or CNT values in the program to 0000; or to reset the preset value:

1. The TMR and CNT current values will be monitored using the procedure beginning on page 3-39.
2. Push the 8/SET key for time up or count up (reset to 0000); or push the 9/RST key to reset to a preset value.

Key operation	Display	Note
CE, CODE CONST, CODE CONST, 1, MNTR	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00004 T 0975 001	The current value of TMR001 is monitored.
8/SET	00004 T 0000 001	Causes it to time up.
STEP (+)	00004 C 0437 002	
9/RST	00004 C 0900 002	The current value of CNT002 is preset to 0900.

Fig. 54, Setting TMR and CNT

It is not possible to set TMR and CNT when the TMR counter input is OFF; or when the CNT reset input is reset.

Changing Current Register Value

In the CHANGE mode it is possible to convert the current value of any register into hexadecimal, octal, decimal, or binary; if the preset value change coil (7365) is ON.

Key operation	Display	Note
	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00000 HEX 23 ⊐001	Data memory ⊐001 is monitored
4 5	00000 HEX 45 ⊐001	The current value (23) is changed to 45.
ENT	00000 HEX 45 ⊐001	The revised current value is written in the memory.
STEP (+) CONV CONV CONV	00000 □■■□□□□□ ⊐002	Changed to the bit pattern.
1 0 1 1 0 0 ENT	00000 □□■□■■□□ ⊐002	Changed to 00101100.

Changing Current TMR, CNT, and MD Preset Values

The following describes how to change the preset values while the PC is running. EXTREME CAUTION IS ADVISED !!!

Key operation	Display	Note
[CLR] [TMR] [1] [SRCH] [INTR]	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 01234 T 0056 TMR 001	TMR001 is searched.
[STEP (+)]	01235 0100	The preset value 0100 is read after advancing address.
[0]	01235 0000	press the key several times to set the data display section to 0.
[2] [0] [0]	01235 0200	Enter the new preset value 0200.
[ENT]	01235 0200	The new preset value is stored in the memory. The display does not change.

Fig. 55, Changing TMR Preset Value

The new preset value will be valid immediately after entry.

Changing Constant

The following describes how to change the value of a constant while the PC is running. EXTREME CAUTION IS ADVISED !!!

Key operation	Display	Note
[CLR] [FUN] [CODE CONST] [1] [0] [SRCH] [MNTR]	7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 00074 Fc10 ADD	Fc10 is searched.
[STEP (+)] [STEP (+)]	00076 14	The BCD constant is read by advancing address.
[0]	00076 00	The key is pressed several times to set the display section to 0.
[1] [5]	00076 15	The new BCD constant is entered.
[ENT]	00076 15	The new BCD constnat is stored in the memory. The display does not change.

Fig. 56, Changing Constant

Confirm the instruction in front of and behind the change, before writing it into memory.

CHAPTER 4 - MEMORY MAPS

Data Memory Address Map

Input and Output Relays 0000 ~ 0777

								Addresses Byte	Absolute
0007	0006	0005	0004	0003	0002	0001	0000	].000	000
0017	0016	0015	0014	0013	0012	0011	0010	].001	001
0027	0026	0025	0024	0023	0022	0021	0020	].002	002
0037	0036	0035	0034	0033	0032	0031	0030	].003	003
0047	0046	0045	0044	0043	0042	0041	0040	].004	004
0057	0056	0055	0054	0053	0052	0051	0050	].005	005
0067	0066	0065	0064	0063	0062	0061	0060	].006	006
0077	0076	0075	0074	0073	0072	0071	0007	].007	007
0107	0106	0105	0104	0103	0102	0101	0010	].010	010
0117	0116	0115	0114	0113	0112	0111	0110	].011	011
0127	0126	0125	0124	0123	0122	0121	0120	].012	012
•									
•									
•									
0677	0676	0675	0674	0673	0672	0671	0670	].067	067
0707	0706	0705	0704	0703	0702	0701	0700	].070	070
0717	0716	0715	0714	0713	0712	0711	0710	].071	071
0727	0726	0725	0724	0723	0722	0721	0720	].072	072
0737	0736	0735	0734	0733	0732	0731	0730	].073	073
0747	0746	0745	0744	0743	0742	0741	0740	].074	074
0757	0756	0755	0754	0753	0752	0751	0750	].075	075
0767	0766	0765	0764	0763	0762	0761	0760	].076	076
0777	0776	0775	0774	0773	0772	0771	0770	].077	077

Example:].036 represents the byte 0360 to 0367

Link Relays 2000 ~ 2777

								Addresses Byte	Absolute
2007	2006	2005	2004	2003	2002	2001	2000	].200	100
2017	2016	2015	2014	2013	2012	2011	2010	].201	101
2027	2026	2025	2024	2023	2022	2021	2020	].202	102
2037	2036	2035	2034	2033	2032	2031	2030	].203	103
2047	2046	2045	2044	2043	2042	2041	2040	].204	104
2057	2056	2055	2054	2053	2052	2051	2050	].205	105
2067	2066	2065	2064	2063	2062	2061	2060	].206	106
2077	2076	2075	2074	2073	2072	2071	2070	].207	107
2107	2106	2105	2104	2103	2102	2101	2010	].210	110
2117	2116	2115	2114	2113	2112	2111	2110	].211	111
2127	2126	2125	2124	2123	2122	2121	2120	].212	112
.									
.									
.									
2677	2676	2675	2674	2673	2672	2671	2670	].267	167
2707	2706	2705	2704	2703	2702	2701	2700	].270	170
2717	2716	2715	2714	2713	2712	2711	2710	].271	171
2727	2726	2725	2724	2723	2722	2721	2720	].272	172
2737	2736	2735	2734	2733	2732	2731	2730	].273	173
2747	2746	2745	2744	2743	2742	2741	2740	].274	174
2757	2756	2755	2754	2753	2752	2751	2750	].275	175
2767	2766	2765	2764	2763	2762	2761	2760	].276	176
2777	2776	2775	2774	2773	2772	2771	2770	].277	177

Example:].232 represents the address 2320 to 2327

Internal Relays 4000 ~ 4777

								Addresses Byte	Absolute
4007	4006	4005	4004	4003	4002	4001	4000	].400	200
4017	4016	4015	4014	4013	4012	4011	4010	].401	201
4027	4026	4025	4024	4023	4022	4021	4020	].402	202
4037	4036	4035	4034	4033	4032	4031	4030	].403	203
4047	4046	4045	4044	4043	4042	4041	4040	].404	204
4057	4056	4055	4054	4053	4052	4051	4050	].405	205
4067	4066	4065	4064	4063	4062	4061	4060	].406	206
4077	4076	4075	4074	4073	4072	4071	4070	].407	207
4107	4106	4105	4104	4103	4102	4101	4010	].410	210
4117	4116	4115	4114	4113	4112	4111	4110	].411	211
4127	4126	4125	4124	4123	4122	4121	4120	].412	212
•									
•									
•									
4677	4676	4675	4674	4673	4672	4671	4670	].467	267
4707	4706	4705	4704	4703	4702	4701	4700	].470	270
4717	4716	4715	4714	4713	4712	4711	4710	].471	271
4727	4726	4725	4724	4723	4722	4721	4720	].472	272
4737	4736	4735	4734	4733	4732	4731	4730	].473	273
4747	4746	4745	4744	4743	4742	4741	4740	].474	274
4757	4756	4755	4754	4753	4752	4751	4750	].475	275
4767	4766	4765	4764	4763	4762	4761	4760	].476	276
4777	4776	4775	4774	4773	4772	4771	4770	].477	277

Example:].455 represents the address 4550 to 4557

Latched Relays 7000 ~ 7377

									Addresses Byte		Absolute
7007	7006	7005	7004	7003	7002	7001	7000		].700		300
7017	7016	7015	7014	7013	7012	7011	7010		].701		301
7027	7026	7025	7024	7023	7022	7021	7020		].702		302
7037	7036	7035	7034	7033	7032	7031	7030		].703		303
7047	7046	7045	7044	7043	7042	7041	7040		].704		304
7057	7056	7055	7054	7053	7052	7051	7050		].705		305
7067	7066	7065	7064	7063	7062	7061	7060		].706		306
7077	7076	7075	7074	7073	7072	7071	7070		].707		307
7107	7106	7105	7104	7103	7102	7101	7010		].710		310
7117	7116	7115	7114	7113	7112	7111	7110		].711		311
7127	7126	7125	7124	7123	7122	7121	7120		].712		312
7267	7266	7265	7264	7263	7262	7261	7260		].726		326
7277	7276	7275	7274	7273	7272	7271	7270		].776		327
7307	7306	7305	7304	7303	7302	7301	7300		].730		330
7317	7316	7315	7314	7313	7312	7311	7310		].731		331
7327	7326	7325	7324	7323	7322	7321	7320		].732		332
7337	7336	7335	7334	7333	7332	7331	7330		].733		333
7347	7346	7345	7344	7343	7342	7341	7340	*	].734	*	334
7357	7356	7355	7354	7353	7352	7351	7350	*	].735	*	335
7367	7366	7365	7364	7363	7362	7361	7360	*	].736	*	336
7377	7376	7375	7374	7373	7372	7371	7370	*	].737	*	337

Example:].700 represents the address 7000 to 7007

* = Latched relay and special register areas (shown on the following page)

Data Memory Address Map (Cont)

Latched Relay Special Area

7340	Special area in which the error code is
7341	represented in the byte address].734
7342	
7343	
7344	
7345	
7346	
7347	
7350	Internal use by the PC only. Can not be
7351	programmed.
7352	
7353	
7354	Non-carry flag
7355	Error flag
7356	Carry flag
7357	Zero flag
7360	0.1 second clock
7361	
7362	
7363	1 second clock
7364	Preset value change coil
7365	Ever-OFF contact
7366	Zero cross switch
7367	
7370	Parity error
7371	CPU failure
7372	Battery failure
7373	I/O failure (internal to controller)
7374	Option module failure
7375	
7376	ROM failure
7377	Power supply failure

TMR, CNT, MD Area

000	b.000	040	b.100	100	b.200	140	b.300
	b.001		b.101		b.201		b.301
001	b.002	041	b.102	101	b.202	141	b.302
	b.003		b.103		b.203		b.303
002	b.004	042	b.104	102	b.204	142	b.304
	b.005		b.105		b.205		b.305
003	b.006	043	b.106	103	b.206	143	b.306
	b.007		b.107		b.207		b.307
004	b.010	044	b.110	104	b.210	144	b.310
	b.011		b.111		b.211		b.311
005	b.012	045	b.112	105	b.212	145	b.312
	b.013		b.113		b.213		b.313
006	b.014	046	b.114	106	b.214	146	b.314
	b.015		b.115		b.215		b.315
007	b.016	047	b.116	107	b.216	147	b.316
	b.017		b.117		b.217		b.317

•
•
•

030	b.060	070	b.160	130	b.260	170	b.360
	b.061		b.161		b.261		b.361
031	b.062	071	b.162	131	b.262	171	b.362
	b.063		b.163		b.263		b.363
032	b.064	072	b.164	132	b.264	172	b.364
	b.065		b.165		b.265		b.365
033	b.066	073	b.166	133	b.266	173	b.366
	b.067		b.167		b.267		b.367
034	b.070	074	b.170	134	b.270	174	b.370
	b.071		b.171		b.271		b.371
035	b.072	075	b.172	135	b.272	175	b.372
	b.073		b.173		b.273		b.373
036	b.074	076	b.174	136	b.274	176	b.374
	b.075		b.175		b.275		b.375
037	b.076	077	b.176	137	b.276	177	b.376
	b.077		b.177		b.277		b.377

The current value for the TMR, CNT, or MD 073 is stored in the two bytes b.166 and b.167

Registers 9000 ~ 9377

9007	9006	9005	9004	9003	9002	9001	9000
9017	9016	9015	9014	9013	9012	9011	9010
9027	9026	9025	9024	9023	9022	9021	9020
9037	9036	9035	9034	9033	9032	9031	9030
9047	9046	9045	9044	9043	9042	9041	9040
9057	9056	9055	9054	9053	9052	9051	9050
9067	9066	9065	9064	9063	9062	9061	9060
9077	9076	9075	9074	9073	9072	9071	9070
9107	9106	9105	9104	9103	9102	9101	9010
9117	9116	9115	9114	9113	9112	9111	9110
9127	9126	9125	9124	9123	9122	9121	9120

•

•

•

9267	9266	9265	9264	9263	9262	9261	9260
9277	9276	9275	9274	9273	9272	9271	9270
9307	9306	9305	9304	9303	9302	9301	9300
9317	9316	9315	9314	9313	9312	9311	9310
9327	9326	9325	9324	9323	9322	9321	9320
9337	9336	9335	9334	9333	9332	9331	9330
9347	9346	9345	9344	9343	9342	9341	9340
9357	9356	9355	9354	9353	9352	9351	9350
9367	9366	9365	9364	9363	9362	9361	9360
9377	9376	9375	9374	9373	9372	9371	9370

File Registers 30000 ~ 33777

30007 to 30377	30006 to 30376	30005 to 30375	30004 to 30374	30003 to 30373	30002 to 30372	30001 to 30371	30000 to 30370
30407 to 30777	30406 to 30776	30405 to 30775	30404 to 30774	30403 to 30773	30402 to 30772	30401 to 30771	30400 to 30770
31007 to 31377	31006 to 31376	31005 to 31375	31004 to 31374	31003 to 31373	31002 to 31372	31001 to 31371	31000 to 31370
31407 to 31777	31406 to 31776	31405 to 31775	31404 to 31774	31403 to 31773	31402 to 31772	31401 to 31771	31400 to 31770
32007 to 32377	32006 to 32376	32005 to 32375	32004 to 32374	32003 to 32373	32002 to 32372	32001 to 32371	32000 to 32370
32407 to 32777	32406 to 32776	32405 to 32775	32404 to 32774	32403 to 32773	32402 to 32772	32401 to 32771	32400 to 32770
33007 to 36777	33006 to 36776	33005 to 36775	33004 to 36774	33003 to 36773	33002 to 36772	33001 to 36771	33000 to 36770
37007 to 33377	37006 to 33376	37005 to 33375	37004 to 33374	37003 to 33373	37002 to 33372	37001 to 33371	37000 to 33370
33407 to 33777	33406 to 33776	33405 to 33775	33404 to 33774	33403 to 33773	33402 to 33772	33401 to 33771	33400 to 33770

File register capacity:

30000 ~ 33777 when configured for 2KB in system memory #205

The system memory consists of 256 bytes (000 ~ 377), and is backed up by batteries.

000 ~ 177 is used by the CPU. Although the user can not write into it, it can be read.

200 ~ 377 is for the user.

Area not open to user		Area open to user	
000	100	200	300
001	101	201	301
002	102	202	302
•		•	
•		•	
•		•	
075	175	275	375
076	176	276	376
077	177	277	377

1. Area in Which Controller Functions are Set

ADDRESS	ITEM	DESCRIPTION
200	Latched relay area	Used to increase or decrease the latched relay area size. The entry must be an octal, absolute address. The size is initially set to 300.

Relay	Relay Address	Byte Address	Absolute Address
I/O Relay	0000	].000	000
	~	~	~
	0777	].077	077
Link Relay	2000	].200	100
	~	~	~
	2777	].277	177
Internal Relay	4000	].400	200
	~	~	~
	4777	].477	277
Latched Relay (not special relay and special register areas	7000	].700	300
	~	~	~
	7337	].733	333

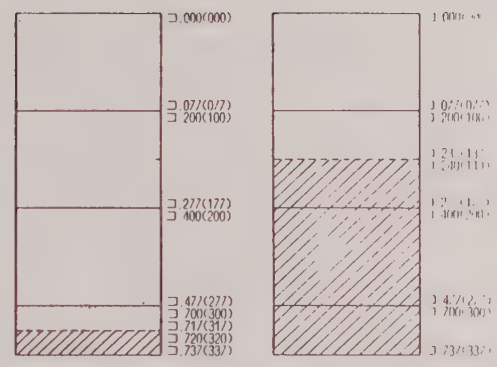
Entered in increments of eight (one byte)

Area in Which Controller Functions are Set (Cont)

ADDRESS	ITEM	DESCRIPTION
201	TMR Reset Condition	Power failure action is set by entering an octal number as shown below: 000 = Reset at power failure 001 = State retained at power failure
202	CNT Reset Condition	Reset condition specified for F-60 or F-62. The number is octal. 000 = Reset with ON signal (default) 001 = Reset with OFF signal
203	Output Retension Address	The absolute address is specified by an octal number for the output relay that retains the output state at HALT. The default setting is 000. All outputs that follow].000 will retain their outputs if the power fails.
204	Program Memory Size	The size is set by entering one of the octal numbers below: 000 = 2.5KW (5KB) 001 = 3.5KW (7KB) (default)
205	File Register Size	The size is set by entering one of the octal numbers below: 000 = not used (default) 001 = 2KB

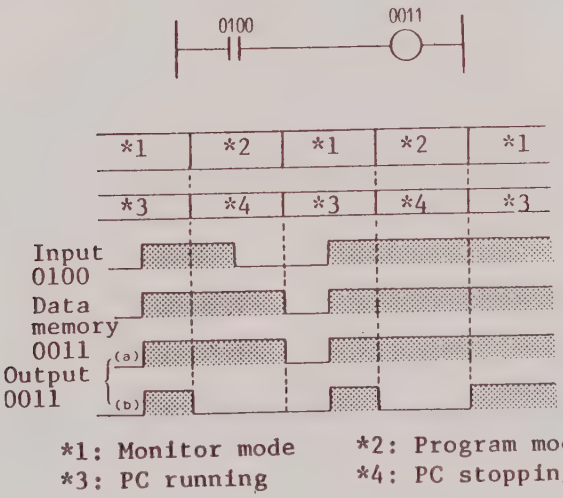
Note 1: After 202 counter reset condition is changed, the reset condition of F-60 and F-62 will also change.

Note 2: After the latched relay area size has been changed, the area that is retained during a power failure is shown in the shaded area.



#200 set to 320 #200 set to 140

Note 3: Although it is possible to set 203 as high as].737, any address higher than].201 will not work.



*1: Monitor mode *2: Program mode
*3: PC running *4: PC stopping

a. When 0011 is within the output retention area.
b. When 0011 is out of the output retention area.

CAUTION

When the PC stops because an error is detected (self-diagnostic), it may not be possible to set the output unit to OFF. Outputs that need to be turned OFF when an error occurs should be wired to the HALT circuit.

2. Error Code Storage Area

210 ~ 217	Error code as a result of self-diagnosis	The error is stored in the area according to its type.

These eight registers allow the storage of up to eight errors. When more than eight errors occur, the error entered first is cancelled.

Note 1: As the error code is not cleared after recovery from the error, 00 must be written by the programmer.

3. Optional Modules Area

220	Remote I/O setup error	30 (BCD) will be written if an error exists
	Remote I/O communication error	31 (BCD) will be written if an error exists
211	Station number of the remote I/O slave unit in error	The station number with the error will send its identification number to this address

Program Memory and File Register

1. Program Memory
The program memory is the area in which the user program is stored. As the PC begins to operate, the program is read sequentially from the instruction in the beginning address to the END command.

The PC 0185 basic configuration is 8KB RAM. A 1KB area is used for data memory and system memory. The remaining 7KB is the program memory. The reader must understand that we speak of 7KB (Bytes) and 3.5KW (Words), as being equal. (16 bits per word)

2. File Register

File Register Capacity Setup

The part of the program memory that is used for data memory is the file register area. For example, if 5KB (2.5KW) of the default 7KB is limited to the program area, the remaining 2KB can be used for the file register area.

The capacity is set using the system memory 204 and 205 addresses:

Program Memory Size (System Memory 204)	File Register Size (System Memory 205)
000 = 2.5KW	001 = 2KB (address 30000 ~ 33777)
001 = 3.5KW	000 = Not Used

Note 1: If the program memory and file register sizes are not set in the above relationship the programmer will not accept them.

Instructions That Can Use the File Registers

The file register is byte designated by the following commands:

- F-05 Demultiplex
- F-06 Multiplex
- F-70 File
- F-71 Constant
- F-72 Demultiplex
- F-73 Multiplex

Since the file register can not be directly specified by instructions like F-10, you must transfer it to normal data areas before operation. (].000, b.xxx, or 9xxx)

3. Write from the Programmer

It is possible to write data directly into the file registers from the programmer. This is accomplished if the preset value change coil (7365) is ON .

Programmer Warning

The PC 0185 and the PP 0110 form a powerful process control tool. Together, they provide you with a high degree of functionality and flexibility. The output responses must be as flexible as possible and completely user definable to allow you complete freedom in designing your control application.

You must, however, be made aware that there are two conditions under which the I/O points will remain ON during a failure:

1. I/O Communications Failure

   If there is a failure in I/O communications while the PC is running the user program, the I/O points that were active before the communications failure may remain active. The points that were not active before the communications failure will stay inactive. Your application must take these possibilities into consideration. The master relay as shown on page 2-3 may be of use.

2. Power Failure

   If the system looses power while the PC is running the user program, the I/O points that were active before the power failure will remain active. Those points that were not active before the power failure will stay inactive. Your application must use the HALT command to latch the critical outputs during power failures; or to reset (turn OFF) all of the outputs.

Also, deleting a coil from the user program after the logic has been solved, even one time, will NOT turn that output point OFF.

1. Suppose that the controller instructs the output unit to turn ON an output point. Then, when the point is in its ON state you exit the run mode, delete the coil from the program, and once again run your program. The I/O acts on its points at the direction of the controller. As far as the controller is concerned the deleted I/O point does not exist. The I/O held all its previous states; and because no further action on that deleted output point was requested by the controller, that point will stay ON. To avoid this situation turn OFF the coil while the controller is running, exit the run mode, delete the coil, and re-enter the run mode. Now the controller has instructed the I/O to turn OFF that point. Power cycling the controller will also delete the coil.

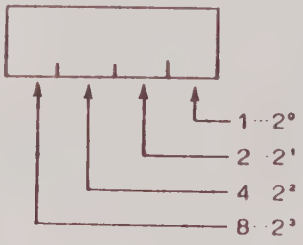
[1] Number Systems

Binary Code

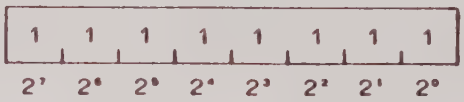
For logic processing only 0 (OFF) and 1 (ON) are used to represent numerical values. This is binary notation. With decimal notation a digit is carried after a number increases from 0 to 9. With binary notation a digit is carried after a number increases from 0 to 1. A binary 10 is used to represent the decimal number 2.

Decimal number	0	1	2*	3	4*	5	6	7	8*
Binary number	0	1	10	11	100	101	110	111	1000

The carry takes place at the point indicated with an *. Each digit has the weight shown below:



Each digit of a binary number is called a bit. Registers in the PC 0185 use eight bits. Eight bits are equal to one byte. Assume all eight bits (one byte) are 1's.



When the total weight of the bits are calculated, we find a value equal to 255 can be represented. Therefore, eight bits can represent any number between 0 and 255.

BCD (Binary Coded Decimal)

With a decimal number, a carry occurs every time the count exceeds 9. That carry, when applied to a binary number, is BCD.

Decimal	Binary	BCD	
0	0	0	
1	1	1	
2	10	10	
3	11	11	
4	100	100	
5	101	101	
6	110	110	
7	111	111	
8	1000	1000	
9	1001	1001	Carry
10	1010	1 0000	
11	1011	1 0001	
12	1100	1 0010	
≈	≈	≈	
99	1100011	1001 1001	

Fig. 57, BCD Code

With the BCD notation, each group of four bits is separated. No combination above 1001 (decimal 9) is permitted. Instead the extra bit(s) are carried to a higher group of four bits.

Octal and Hex

In the PC all numbers are treated in the BCD or binary form. However, the PC permits programming and monitoring using decimal numbers. This is accomplished by using a converter inside the programmer. It is often preferable to use a number system where the bit pattern can be recognized without the use of converters. Octal and Hex are two such numbering systems.

Octal

A carry occurs when a number increases over 7. So that to represent 8 (decimal) we would require the number to be 10 in octal.

Decimal	Binary	Octal	
0	0	0	
1	1	1	
2	10	2	
3	11	3	
4	100	4	
5	101	5	
6	110	6	
7	111	7	
8	1000	10	Carry
9	1001	11	
10	1010	12	
11	1011	13	
12	1100	14	
13	1101	15	
14	1110	16	
15	1111	17	
16	10000	20	Carry
≈	≈	≈	
62	111110	76	
63	111111	77	
64	1000000	100	Carry
65	1000001	101	

Fig. 59, Octal Code

Octal Number vs Binary Number

0 through 7 are represented by three binary digits and a carry occurs when it changes from 111 to 1000. With an octal number, a single digit can represent a number 0 thru 7, and the carry occurs when the number increases from 7 to 10. Because a carry occurs immediately after 7 for both the binary and octal number, a binary digit separated into three digits can be used to represent one number in octal.

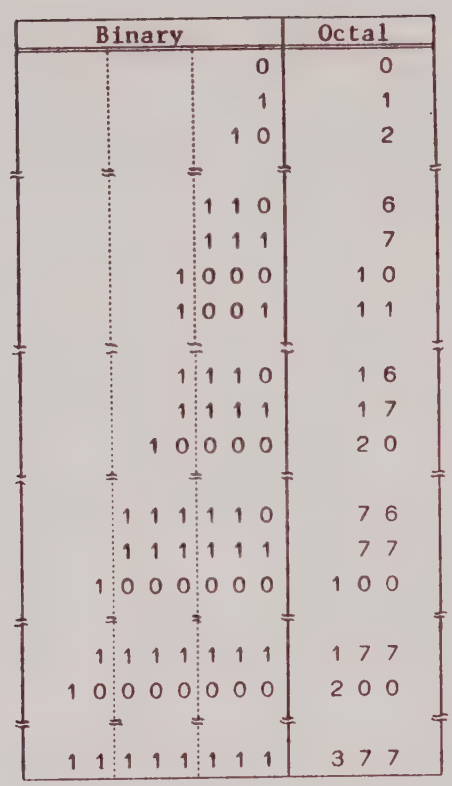


Fig. 60, Octal Code

Since the registers contain eight bits it can be used to represent any number between 0 and 377 in octal (0 to 255 in decimal).

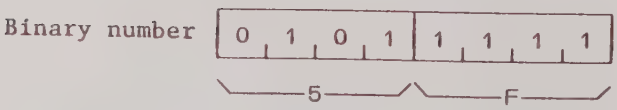
Data memory, system memory, and program memory addresses are expressed in octal numbers.

Hexadecimal (Hex) Notation

A carry occurs when the number increases from F (decimal 15) to 10. Hex numbers can be represented by groups of four bits; in the same way that octal can be represented by groups of three bits.

Dec.	Binary	Octal	Hex.
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	1 0
≈	≈	≈	≈
31	11111	37	1 F
32	100000	40	2 0
≈	≈	≈	≈
255	11111111	377	F F

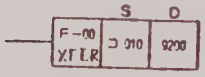
Fig. 57, Hex Code



[2] Source and Destination

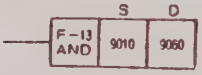
With the data processing instruction data memory is handled in bytes. The register that contains the data before the operation is called the source (S). The register in which the result is stored is called the destination (D).

(Ex-1)



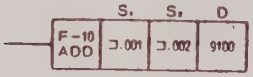
The contents of J.010(S) are transferred to 9200(D).

(Ex-2)



The contents of 9010(S) are ANDed with the contents of 9060(D) and its result is stored in 9060(D).

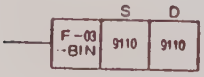
(Ex-3)



The contents of J.001(S_1) are added with the contents of .002(S_2) and its result is stored in 9100(D).

The contents of the register on the source side does not change after the execution. It is possible to make the destination register the same as the source register. In this way it is possible to over-write data.

(Ex-4)



The contents of BCD 2-digit 9110(S) are converted into a binary code and its result is stored in 9110(D).

[3] Application Instruction Stack Register
With the PC 0185 the TMR, CNT, MD, and the application instructions (F-00 to F-73) are processed by a high speed, general purpose CPU. For these instructions, contents of the accumulator and stack register are sent via the data bus to the CPU.

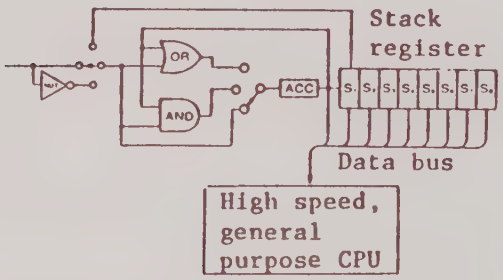
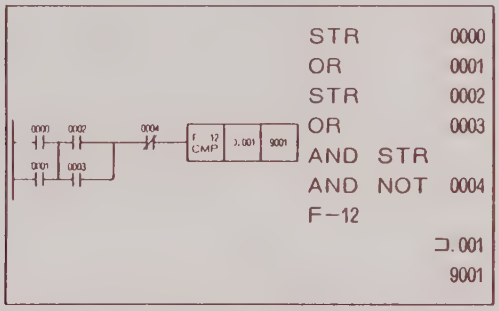


Fig. 62, CPU Route

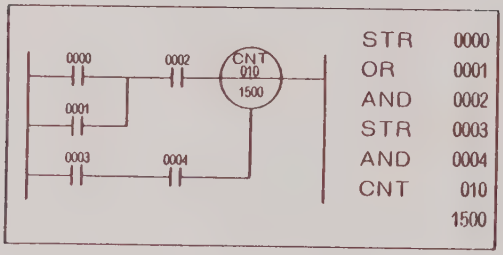
Example:

Application instructions, except CNT, MD, F-80, F-61, and F-62 are executed after receiving the contents of the accumulator (after the operational conditions are met).



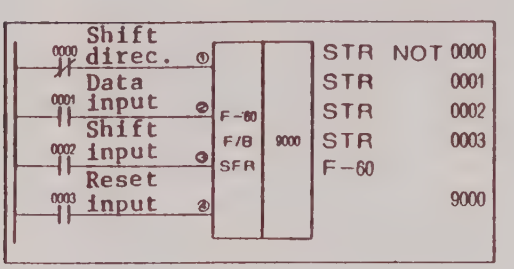
Instruction	Accumulator ACC	Stack register S,
STR 0000	0000	
OR 0001	0000 / 0001	
STR 0002	0002	0000 / 0001
OR 0003	0002 / 0003	0000 / 0001
AND STR	0000 0002 / 0001 0003	
AND NOT 0004	0000 0002 0004 / 0001 0003	
F-12	Operation when condition is met	

Example of the CNT Instruction



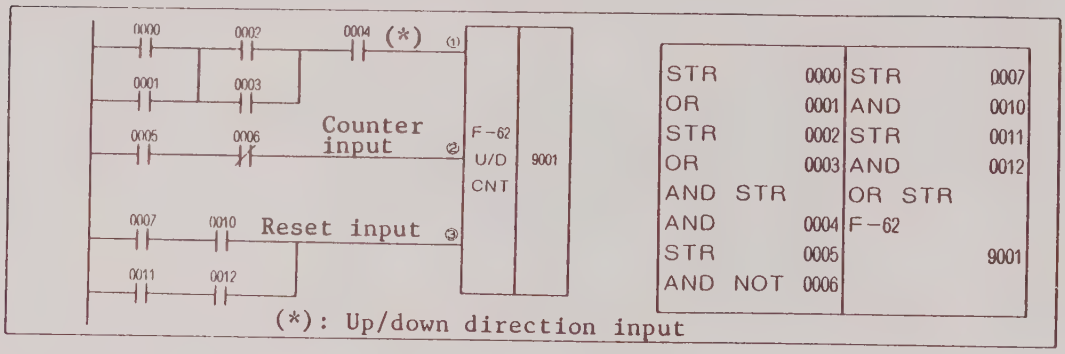
Instruction	Accumulator ACC	Stack register S,
STR 0000	0000	
OR 0001	0000 / 0001	
AND 0002	0000 0002 / 0001	
STR 0003	0003	0000 0002 / 0001
AND 0004	0003 0004	0000 0002 / 0001
CNT 010 1500	Reset input	Counter input

With the F-60 instruction the accumulator and stack registers (S1 to S3) become the operational condition.



Instr.	Accumulator ACC	Stack register S1	S2	S3
STR NOT 0000	0000			
STR 0001	0001	0000		
STR 0002	0002	0001	0000	
STR 0003	0003	0002	0001	0000
F-60	Reset input ④	Shift input ③	Data input ②	Shift direction (1)

The contents of the stack register may be a complex serial/parallel circuit.

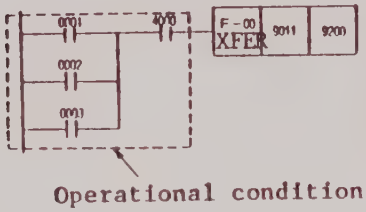


STR 0000	STR 0007
OR 0001	AND 0010
STR 0002	STR 0011
OR 0003	AND 0012
AND STR	OR STR
AND 0004	F-62
STR 0005	9001
AND NOT 0006	

Instruction	Accumulator ACC	Stack register S_1	S_2	S_3
STR 0000	0000			
OR 0001	0000 0001			
STR 0002	0002	0000 0001		
OR 0003	0002 0003	0000 0001		
AND STR	0000 0001 0002 0003			
AND 0004	0000 0001 0002 0003 0004			
STR 0005	0005	0000 0001 0002 0003 0004		
AND NOT 0006	0005 0006	0000 0001 0002 0003 0004		
STR 0007	0007	0005 0006	0000 0001 0002 0003 0004	
AND 0010	0007 0010	0005 0006	0000 0001 0002 0003 0004	
STR 0011	0011	0007 0010	0005 0006	0000 0001 0002 0003 0004
AND 0012	0011 0012	0007 0010	0005 0006	0000 0001 0002 0003 0004
OR STR	0007 0010 0011 0012	0005 0006	0000 0001 0002 0003 0004	
F-62	Reset input ③	Counter input ②	Up/down direction input ①	

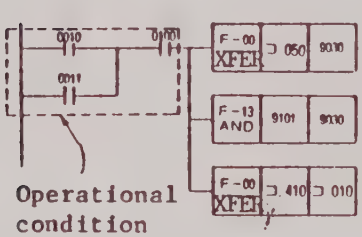
[4] Operational Condition

1. The operational condition of an application instruction may be a complex serial/parallel circuit.



Instruction	
STR	0001
OR	0002
OR	0003
AND	4000
F-00	
	9011
	9200

2. If the operational condition is the same, programming like this may be developed.



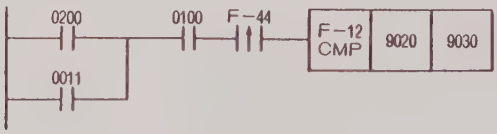
Instruction	
STR	0010
OR	0011
AND	0100
F-00	
	⊐ 050
	9030
F-13	
	9101
	9030
F-00	
	⊐ 410
	⊐ 010

3. There are two kinds of processing for application instructions when the operational condition is satisfied. (F-30, F-31, F-40, F-41, F-42, and F-43 are exceptions)

①	Those which operation takes place at each scan cycle, so long as the operational condition is met.	F-12 (compare register with register) Fc12 (compare register with constant) F-61 (asynchronous shift register)
②	Those which operation takes place only at the first scan cycle that the operational condition is met.	Data processing instructions except the above.

With the instruction of group 2 above, the ON/OFF state of the operational condition in the preceding scan cycle is compared to the ON/OFF state of the operating conditional in the current scan cycle. If the preceding state is OFF and the current state is ON the operation takes place as if the operational condition changed from OFF to ON.

If the operation must be executed only when there is an OFF to ON transition in the operational condition when using the F-12, Fc12, F-61, or F-44 instructions a rise edge differentiate instruction must be used.



4. If the operational condition is not met, no operation takes place and the destination register remains unaffected. The operation is executed only when the operational condition changes from OFF to ON. If the instruction affects the flag, it clears the flag.

[5] Data Processing Instruction and Flag

1. Kinds of Flags

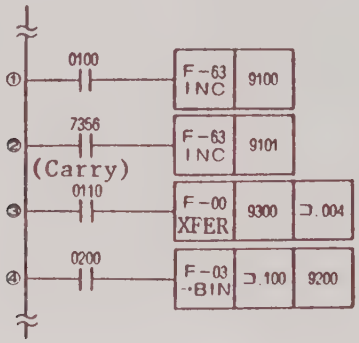
A flag is used to reflect the operational result. There are four kinds of flags for the PC 0185: non-carry, carry, error, zero. They are assigned to four bits in data memory 7354 to 7357.

Non-carry flag	Error flag	Carry flag	Zero flag
7354	7355	7356	7357

2. Instructions That Affect Flags

1	F-03	Convert BCD (2-digit) to binary
2	F-10	Add (BCD) register with register
3	Fc10	Add register with BCD constant
4	F-11	Subtract (BCD) register from register
5	Fc11	Subtract register by BCD constant
6	F-12	Compare registers
7	Fc12	Compare register with constant
8	F-15	Multiply register by register
9	Fc15	Multiply register by BCD constant
10	F-16	Divide register by register
11	Fc16	Divide register by BCD constant
12	F-53	Convert BCD (4-digit) to binary
13	F-60	Shift register bidirectional
14	F-61	Shift register asynchronous
15	F-62	Set BCD up/down counter
16	F-63	Increment counter
17	F-64	Decrement counter

3. Transition of Flag State in Scan Cycle
Flags are cleared prior to processing of the user program; if there are no flag affecting instructions in the preceding steps.

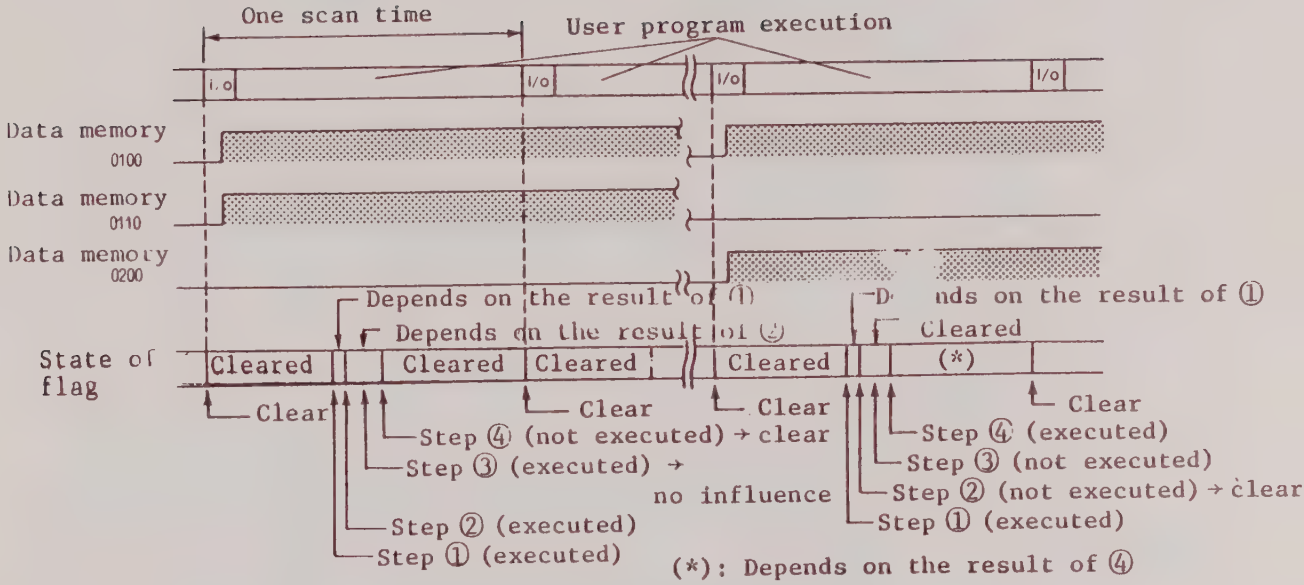


When it processes a flag affecting instruction:
The flag is set according to the operational result of the instruction.

The flag is cleared when the operational condition is not satisfactory for that instruction.

With no flag affecting instruction the state of the flag is not changed, whether the operation took place or not.

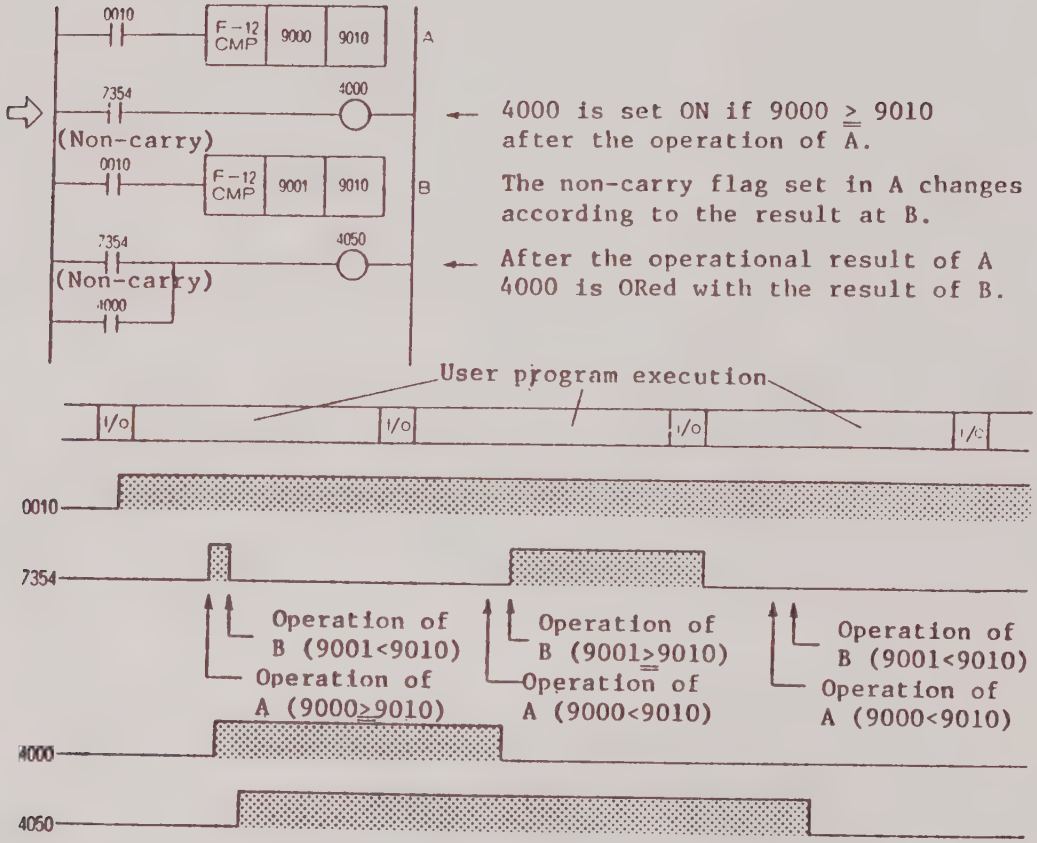
Addr.	Instruction		
01000	STR	0100	①
01001	F-63		
01002		9100	
01003	STR	7356	②
01004	F-63		
01005		9101	
01006	STR	0110	③
01007	F-00		
01010		9300	
01011		⊐.004	
01012	STR	0200	④
01013	F-03		
01014		⊐.100	
01015		9200	



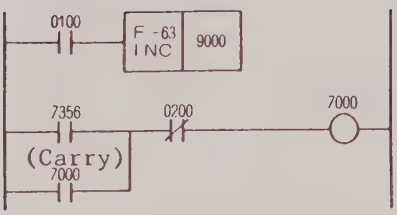
4. How to Retain a Flag

A flag set according to the operational result may be changed or cleared by execution of the flag-affecting instruction in that scan cycle. On the next cycle the flag is cleared before the execution of the user program. To retain the state of the flag, you must write the state of the flag to a coil (such as internal relay) immediately after the instruction that affects the flag. The state of the flag is retained until the following scan.

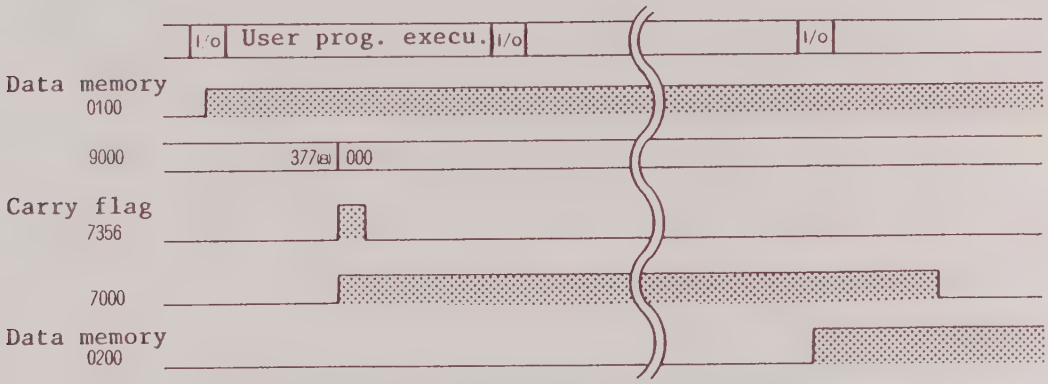
Program that sets 4050 ON when 9000 > or = 9010 OR when 9001 > or = 9010.



When the state of the flag is to be monitored you must store the state of the flag since it will be cleared every cycle.



The state of the carry flag (7356) immediately after the operation of ① can be self-retained until 0200 is set ON.



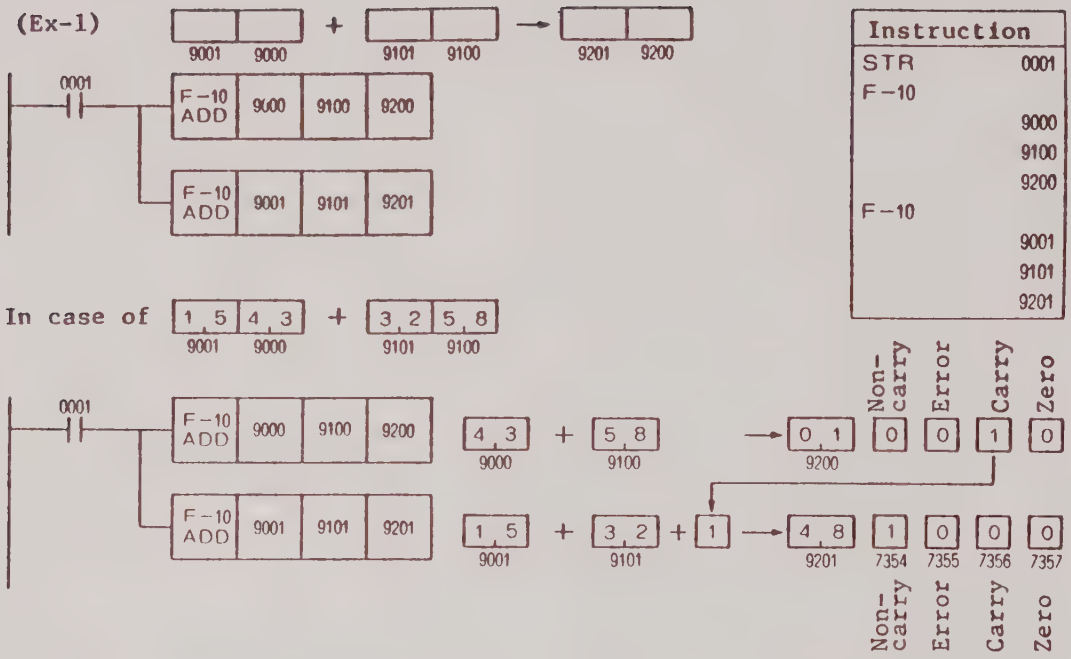
[6] Double Length Operation

The following six instructions operate on more that two bytes.

F-10 add register with register
Fc10 add register with BCD constant
F-11 subtract register from register
Fc11 subtract register by BCD constant
F-12 compare registers
Fc12 Compare register with constant

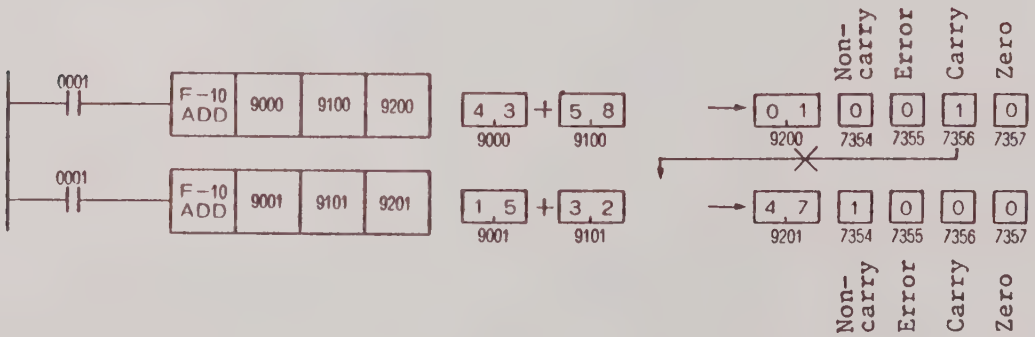
Programming Double Length Operation

In double length operation a carry over or carry down occurs in the lower order digit, and is reflected in processing the next digit. It should be programmed from the low order digit in succession to the operational condition.

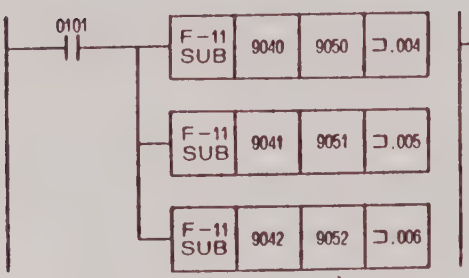


The carry flag in a low order digit is added to the next high order digit during the operation.

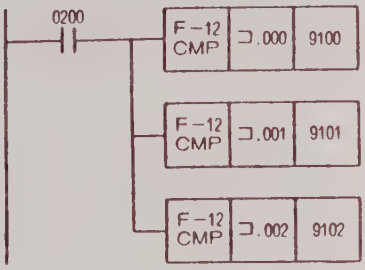
Programming in the following way will NOT perform a double length operation:



Double length operation with more than three digits:



(Subtraction of BCD 6 digits)



(Comparison of three bytes data)

3. Internal Processing During Double Length Operations
For the operation of F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) the instruction immediately after the operational condition takes place without including the state of the flag that precedes it.

If there is a F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instruction encountered during the execution of the common operational condition the following takes place:

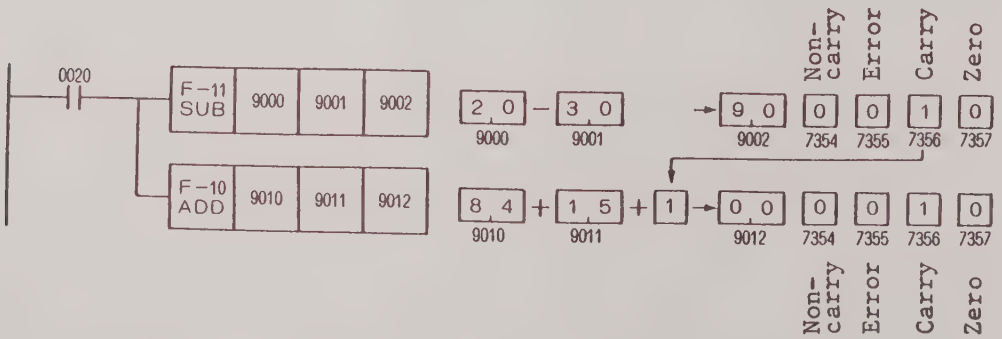
- The state of the carry flag immediately before the operation is used.
- The state of the zero flag immediately before is ANDed with the state of the zero flag immediately after the instruction is executed. If both are 1, the zero flag is set.

F-10 (Fc10)	Adds the state of the carry flag immediately before
F-11 (Fc11)	Subtracts the state of the carry flag immediately before
F-12 (Fc12)	Subtracts the state of the carry flag immediately before

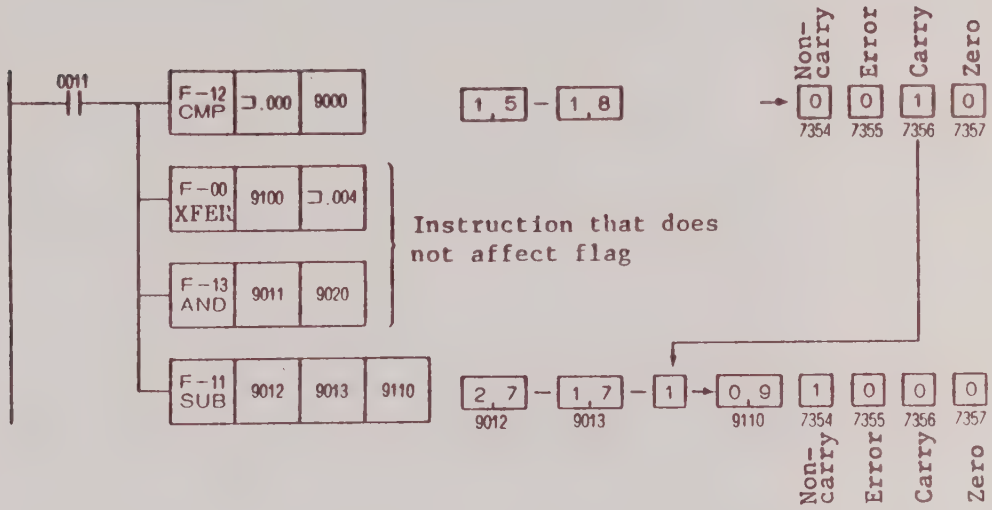
The F-12 (Fc12) instruction performs the operation of S1-S2 or S1-n; and the result is stored in the flag.

4. Cautionary Notes

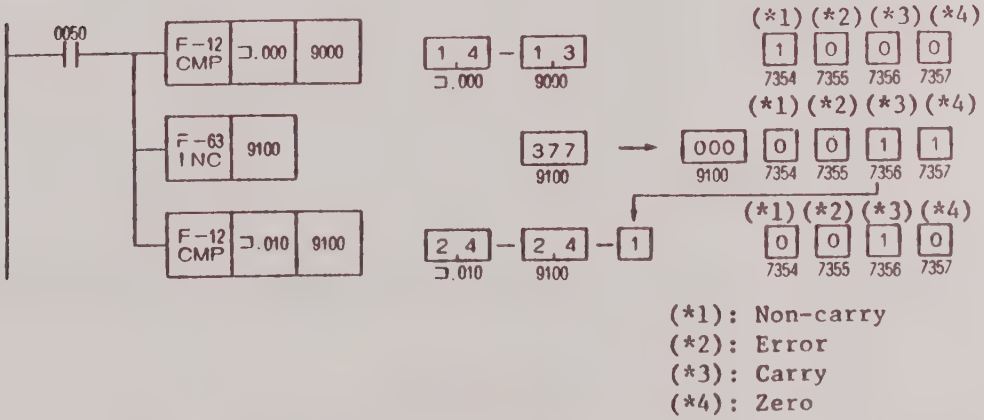
If F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instructions have been programmed to have a common operational condition, be aware that the flag operation occurs at different times for each instruction.



The double length operation takes place even if there is no flag-affecting instruction between the F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instructions.



If there is a flag-affecting instruction between the F-10 (Fc10), F-11 (Fc11), or F-12 (Fc12) instruction, execution takes place including the flag caused by the execution of that instruction.



[7] Using The Relay Area

It's possible to use the relay input area (000 to 0777), link relay area (2000 to 2777), internal relay area (4000 to 4777) and the latched relay area (4000 to 4777) as registers; if a byte is specified, using byte addressing].nnn.

1. There are open relay numbers in each group, and they are arranged successively. The following instructions need to use more than two bytes for source and destination.

F-05	Demultiplex
F-06	Multiplex
F-15	Multiply register by register
Fc15	Multiply register by BCD constant
F-16	Divide register by register
Fc16	Divide register by BCD constant
F-50	Decode 4 to 16
F-51	Encode 16 to 4
F-53	Convert BCD (4-digit) to binary (16-bits)
F-54	Convert binary (16-bit) to BCD (6-digit)
F-61	Shift register asynchronous
F-70	n-bytes batch transfer
F-71	Octal constant batch transfer
F-72	Demultiplex file register
F-73	Multiplex file register

(Ex-1)

(Ex-2)

The 7-byte contents of 9100 ∿ 9106 are transferred to 4 bytes of ⊐.074 ∿ ⊐.077 and 3 bytes of ⊐.200 ∿ ⊐.202.

(Ex-3)

Absolute addressing must be used for 9100 (data pointer).

	Byte address	Absolute address
I/O relay	⊐.000∿⊐.077	000∿077
Link relay	⊐.200∿⊐.277	100∿177
Auxiliary relay	⊐.400∿⊐.477	200∿277
Latched relay	⊐.700∿⊐.737	300∿337

2. Latched Relay Special Area

The four bytes of].734 to].737 are a special area that can't be used for coils or destinations by the user's program.

[8] Data Memory Block and Reference Address

When the data memory is divided into blocks (each block with 256 bytes), the first address is called the reference address.

Reference address	Block	Range
⊐.000	I/O relay, link relay, auxiliary relay, latched relay	⊐.000 ∿ ⊐.737
b.000	TMR and CNT current values and MD information	b.000 ∿ b.377
9000	Register	9000 ∿ 9377
30000	File register ①	30000 ∿ 30377
30400	File register ②	30400 ∿ 30777
31000	File register ③	31000 ∿ 31377
31400	File register ④	31400 ∿ 31777
32000	File register ⑤	32000 ∿ 32377
32400	File register ⑥	32400 ∿ 32777
33000	File register ⑦	33000 ∿ 33377

The block from].000 to].737 is only 224 bytes

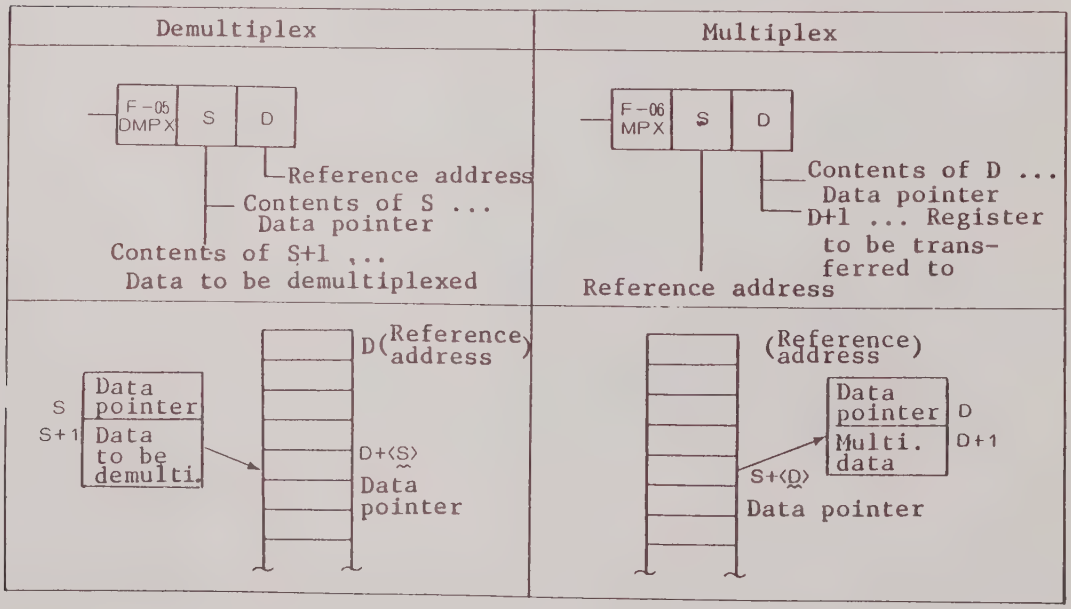
When the file register capacity is set to 2KB, it is possible to use addresses 30000 to 33777.

1. Instruction That Uses the Reference Address

F-05 demultiplex
F-06 multiplex
F-72 demultiplex file register
F-73 multiplex file register

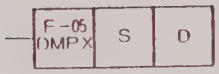
Those instructions transfer data between registers. The address of the register that the data is transferred to is specified by (the reference number plus data pointer).

For F-05 and F-06:



Reference Address

The destination (D) of F-05 and source (S) of F-06 and the first address of each block (].000, 9000, 30000, 30400, 31000,.....37400) are used. Although it's possible to program the use of an address other than the beginning of an address block, Th PC operates assuming the beginning address of a block is the reference address.

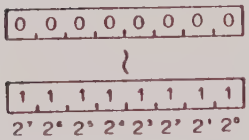


In regard to:

D in setup	Reference address to be operated on
⊐.200	⊐.000
b.110	b 000
9005	9000
30210	30000

Data Pointer

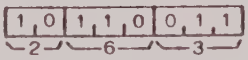
The contents of S of F-05 and D of F-06 become the data pointer. Since S and D are composed of one byte, it is possible to represent any number from 0 to 255.



$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0$
$=128+64+32+16+8+4+2+1$
$=255$

It is possible with "reference address + data pointer" to assign any data memory area within the block for the demultiplexing destination and multiplexing source. The data memory uses octal numbering for byte addressing. This allows direct sensing of the address of a specific register. (If the contents of the data pointer are octal as well.) It is possible to change the demultiplexing destination and the multiplexing source by changing the contents of the data pointer by using F-63 (INC) or specifying an external device.

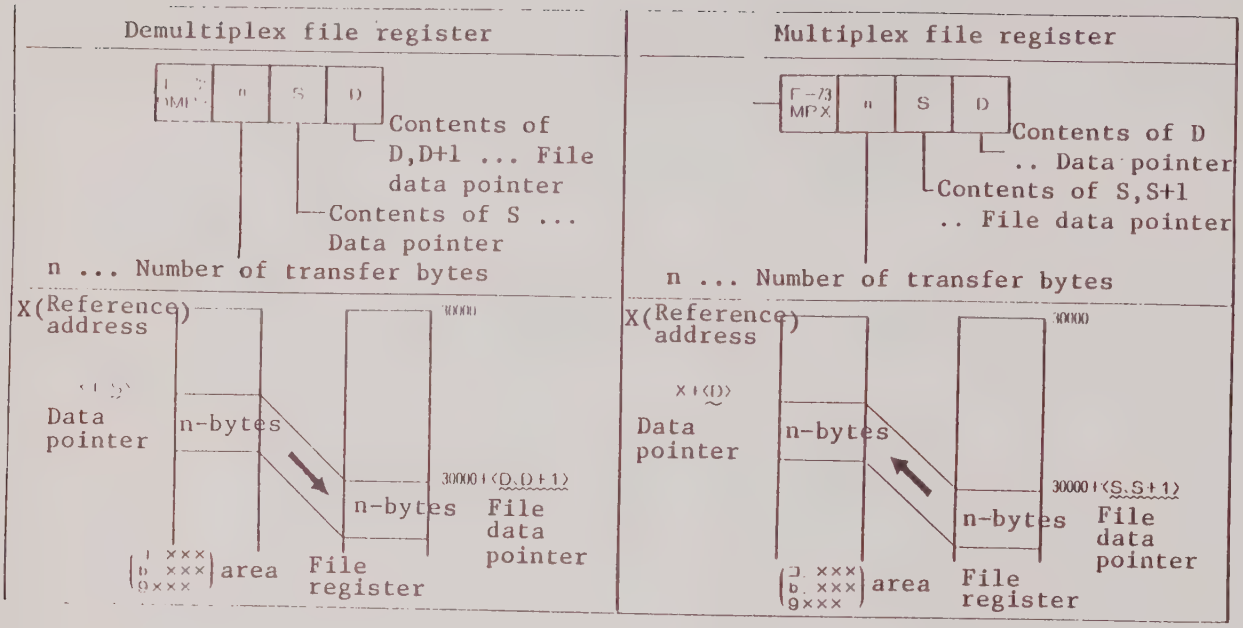
Reference address 9000

Contents of
the data pointer ..... 

179 in the decimal notation

9000 + 263 = 9263 is the demultiplexing destination (multiplexing source).

For the F-72 and F-73



These instructions limit the demultiplexing destination and multiplexing source to file register. They are capable of multiplexing and demultiplexing several bytes in batch operation. The transfer starting address is dependent upon the reference address plus the data pointer for the register side; and 30000 plus file data pointer for the file register side.

The file data pointer by 12 bits in a 2-byte register.

(4095 in decimal notation)

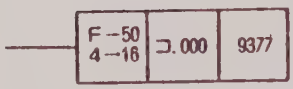
So it is possible to represent any file register between 0 and 33777.

2. Memory Block Size Considerations

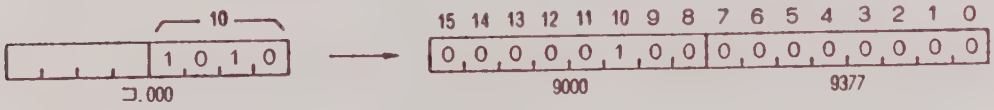
An instruction with a source and destination requiring more than two bytes.

If the S and D will not fit at the bottom of the address block, the least significant byte is entered in the last address for the block and the most significant address is entered at the reference address for the block.

(Ex)



Low order 4 bits of ⊐.000 are assumed to be a binary number and decoded, and its result is written in two bytes having 9377 for its top address

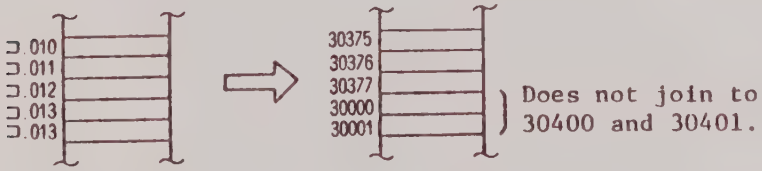


For b-type batch transfer instructions (F-70, F-72, and F-73):

An instruction with a source and destination requiring more than two bytes.

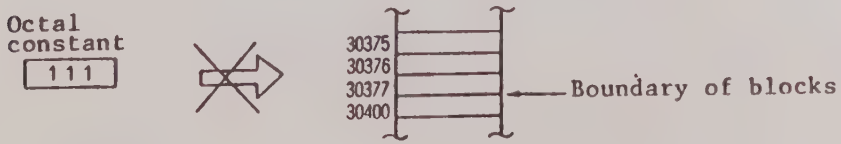
If the S and D will not fit at the bottom of the address block, the least significant byte is entered in the last address for the block and the most significant address is entered at the reference address for the block.

(Ex)

The 5-bytes contents having ⊐.010 for its top address is transferred to 5 bytes having 30375 for its top address.

The octal constant batch transfer instruction does not have the ability to wrap around to the reference address of the address block.

(Ex)

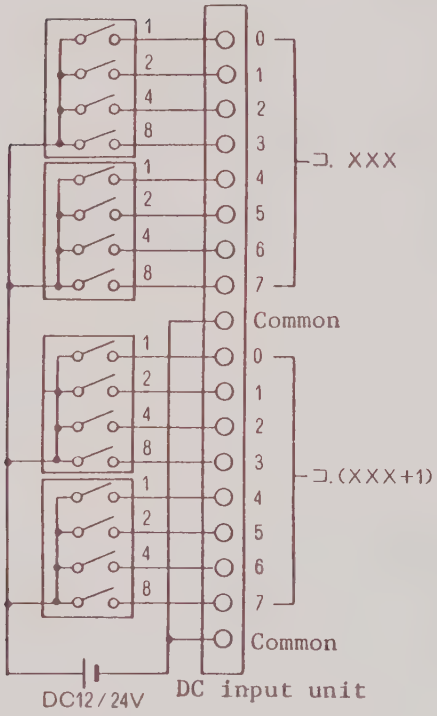
No execution takes place with the program that the octal constant 111 is transferred to the range of 30375 through 30400.



[9] Numerical Signal I/O Method

Some external devices communicate with the PC 0185 by using a numerical signal sensed from a digital switch.

1. Input



These connections read the signal into the data memory I/O on each scan.
The data (2 bytes) can be specified by means of the data processing instruction for one byte of].nnn and another byte of].nnn+1.

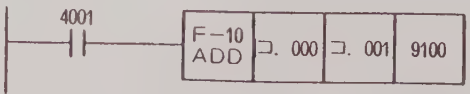
Weight \ No.	0	1	2	3	4	5	6	7	8	9
1		•		•		•		•		•
2			•	•			•	•		
4					•	•	•	•		
8									•	•

• — Dot represents the ON state of the switch.

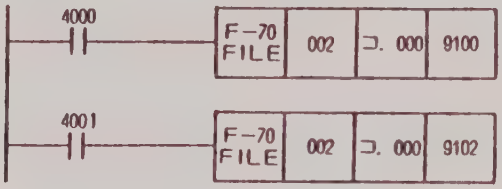
, When a complementary code digital switch is used, the signal should be inverted using the F-09 (INV) instruction.

BCD two digits of].000 is added to BCD two digits of].001 and the result is stored in register 9100. It can then be used with the data processing instruction.

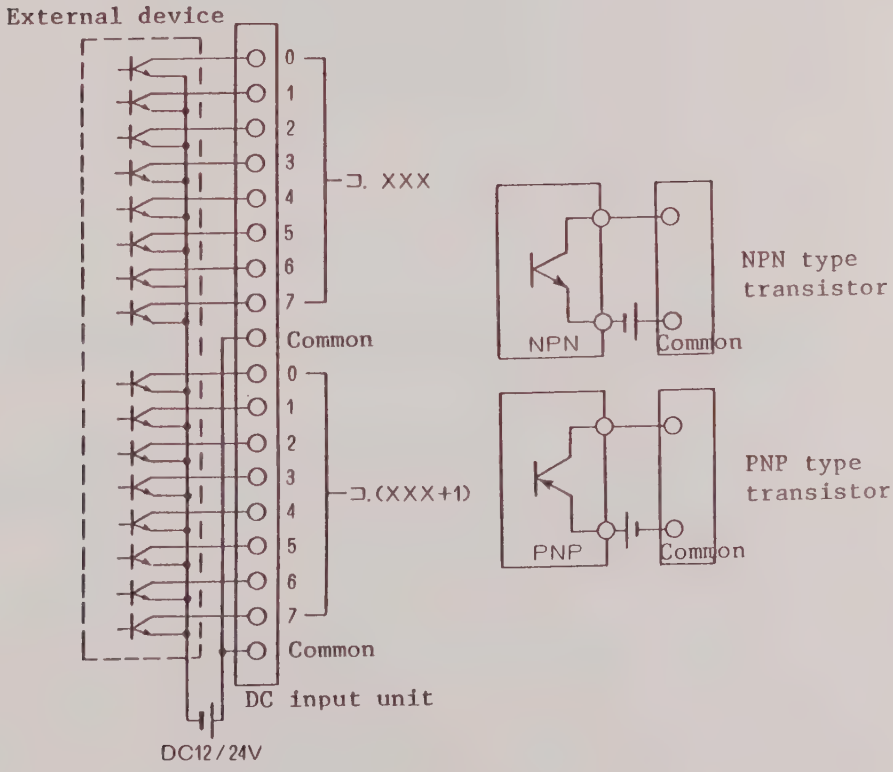
(Ex)



When 4000 is set ON, two bytes (BCD 4 digits) of].000 and].001 are transferred to 9100 and 9101.
When 4001 is set ON, two bytes (BCD 4 digits) of].000 and].001 are transferred to 9102 and 9103.



In the example above, a multiple number of preset values can be read with a pair of digital switches.

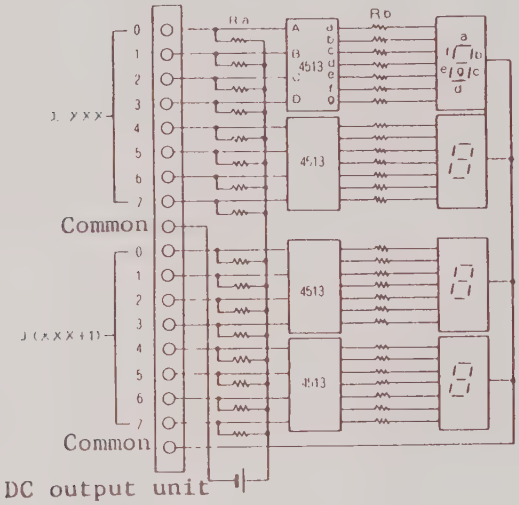


2.Output of Numeric Signal

Display Output Devices

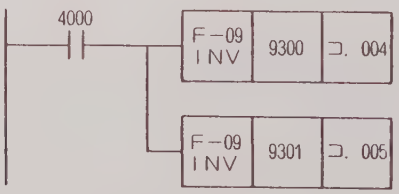
Use the common cathode type of 7 segment display.

Use CMOS MC4513 as the decoder driver.

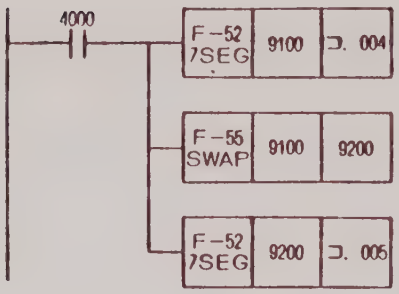


The display logic above is positive. Output data must be inverted (F-09) before being transferred to the I/O relay area of the data memory.

The contents of the register 9300 are inverted and stored in].004 (low order digits). The contents of register 9301 are inverted and stored in].005 (high order digits).



Example for a one byte, 2 digit number:



Convert the low order four bits of register 9100 into the 7 segment data to be output to].004. Swap the high order four bits of the register 9100 with the low order bits and store the result in register 9200. Convert the low order bits in register 9200 into the 7 segment data to be output to].005.

Appendix A - Key Functions in Each Mode

Mode Key(s) Used	PROGRAM	CHANGE	MONITOR
MD, TMR, CNT, FUN, NOT, AND, OR, STR, OUT	For instruction entry		
0 ~ 7	Relay address entry TMR, CNT, and MD address entry Register address entry Program address entry FUN number entry System memory address entry		
	For preset value entry (0000~1999)		
	For Constant entry		
	EL Illumination ON/OFF entry (**0) Buzzer choice entry (**1)		
	Tape recorder oper.		
	Memory clear		
8 SET	FUN number setup For preset value, constant setup		
	Mode change setup Start tape recorder		
	Activation of EL illumination Activation of buzzer		
		Latch relay setup TMR, CNT setup	
9 RST	FUN number reset For preset value, constant reset		
	Memory clear		
	Deactivation of EL illumination Deactivation of buzzer		
		Latch relay reset	
PROG MODE	Memory clear System memory initialization		
CHNG MODE	Change mode (**8/SET)		
MNTR MODE			
CODE CONS	Data memory selection MD output area assignment Application instruction register (constant) assignment		

Key Functions in Each Mode (Cont)

Mode Key(s) Used	PROGRAM	CHANGE	MONITOR
CONV	To convert number systems - hex to octal to decimal to binary		
*	For mode change EL illumination and buzzer ON/OFF choice Memory clear Program check Parity check		
SYS MEM	For system memory setup		
CONT	To continue program and parity check		
REC VERI	Cassette tape operation		
LOAD	Recorder operation		
ENT	Program write	Write preset,value or constant	
	Write System memory Memory clear		
		Change current register value	
INS	Insert instruction		
DEL	Delete instruction		
ADRS	For address setup		
SRCH	Instruction search Data memory search Program check		
MNTR	Instruction read Data memory read System memory read Search		
		Register byte/word monitor mode change	
		Stack/flag read at breakpoint	

Key functions in Each Mode (Cont)

Mode Key(s) Used	PROGRAM	CHANGE	MONITOR
STEP (-)	Instruction read (address minus one) Data memory read (address minus one) System memory read (address minus one) Search in reverse address direction		
STEP (+)	Instruction read (address plus one) Data memory read (address plus one) System memory read (address plus one)		
CE	Clear the last instruction entered		
CLR	All display initialization Parity check For memory Clear Stop program loading Stop recording and verification		

Operation by Functional Groups

	NAME	MODE	OPERATION
1	EL ON/OFF	PROGRAM MONITOR CHANGE	* * 0 { 8 SET ON / 9 RST OFF
2	Buzzer ON/OFF	PROGRAM MONITOR CHANGE	* * 1 { 8 SET ON / 9 RST OFF
3	Mode Change	PROGRAM MONITOR CHANGE	9 9 { PROG MODE 8 SET PROGRAM mode / MNTR MODE 8 SET MONITOR mode / CHNG MODE 8 SET CHANGE mode
4	Clear all memory	PROGRAM	* 0 PROG MODE 8 SET CLR 9 RST 0 ENT
	Clear file		* * PROG MODE 8 SET CLR 9 RST ENT
	Clear Expanded Memory		* * PROG MODE 8 SET CLR 9 RST → 1 ENT 3.5KW
	Initialize System Memory		* * PROG MODE 8 SET CLR 9 RST SYS MEM ENT
5	Program memory addresses	PROGRAM MONITOR CHANGE	CLR ADRS 0 ~ 7 Program memory address
	System memory addresses	PROGRAM MONITOR CHANGE	CLR SYS MEM 0 ~ 7 System memory address
	Data memory addresses		CE CODE CONST ..... 0 ~ 7 Data memory address →0000→000→□000→b000→9000→30000

Operation by Functional Groups (Cont)

	NAME	MODE	OPERATION
6	Program memory read	PROGRAM MONITOR CHANGE	Program memory address setup → MNTR → STEP (−) / STEP (+); or STEP (−) / STEP (+)
	System memory read		System memory address setup → MNTR → STEP (−) / STEP (+); or STEP (−) / STEP (+)
	System memory read value code conversion		System memory read → CONV ..... → BCD to octal, decimal, and to binary (bit pattern)
	Data memory read	MONITOR CHANGE	Data memory address entry → MNTR → STEP (−) / STEP (+); or STEP (−) / STEP (+)
	Data memory read value code conversion		Data memory read → CONV ..... → BCD to octal, decimal, and to binary (bit pattern)
7	Program memory	PROGRAM	..... STR 5 ENT STEP (+) — Instruction entry
	System memory		System memory read → 0 ~ 7 ENT STEP (+) — Numeric entry
8	Parity check	PROGRAM	* * CLR; In the case of repeated check CONT
	Program check		* * SRCH; In the case of repeated check CONT

	NAME	MODE	OPERATION
9	Instruction	PROGRAM MONITOR CHANGE	CLR OUT 1 SRCH MNTR ····· SRCH MNTR Instruction entry Repeated search
	Data memory		Data memory entry → SRCH MNTR ····· SRCH MNTR Repeated search
10	Latched relay	PROGRAM MONITOR CHANGE	Latched relay read → 0 SET / 0 RST
	TMR, CNT		TMR, CNT read → 0 SET / 0 RST
11	Changing preset value in middle of operation	CHANGE	Preset value, constant read → 0 ~ 0 RST ENT New value
12	Changing register value in middle of operation	CHANGE	Register read → 0 ~ 0 RST ENT New value
13	Record	PROGRAM MONITOR CHANGE	CLR REC { 0 0 SET Data memory; 1 0 SET Program memory System memory
14	Verify		CLR VERI { 0 0 SET Data memory; 1 0 SET Program memory System memory
15	Load	PROGRAM	CLR LOAD { 0 0 SET Data memory; 1 0 SET Program memory System memory

Appendix B

Display Message List

ITEM	MESSAGE	MEANING
At Start	SET CU SW TO IN-USE TO USE PC	Set the programmer port to ON
	00000 3.5KW	Memory capacity is set to 3.5KW (7KB) - this can be altered
Error Indication	TIME OUT	Signal does not return from the PC
	XMIT ERROR	Transmission error
	WRONG KEY	Incorrect key press
Enter, insert, delete	BUSY	Enter, insert or delete in progress
Search	IN SRCH	Search is in progress
	NOT FOUND	The instruction searched for was not found after search was completed
Load, record, verify	DATA (0) PROGRAMME (1)	Push the 0 key to load, record or verify data memory - push the 1 key to load, record or verify the program memory.
	PUSH SET KEY	Push the 8/SET key on the programmer.
	MARK	Ready to begin load, record, or verify.
	1-	First load, record, or verify running
	TAPE STOP	Pressing the CLR key while record, load, or verify was in progress
	TAPE ERR	Record, load or verify error
Recording	RECORD	Recording operation
	REC START	Recording will begin
	IN REC	Recording is in progress
	IRC	Non-signal period
Verifying	VERIFY	Verifying operation
	VER STRT	Verifying will begin
	IN VERIFY	Verifying is in progress
	2-1 (OK)	2nd verify in progress; 1st OK
Loading	LOAD	Loading operation
	LOAD STRT	Loading will begin
	IN LOAD	Loading is in progress
	LOAD DONE	Load complete

Display Message List (Cont)

ITEM	MESSAGE	MEANING
Verify the recording	TAPE ERR0	Start bit not found
	TAPE ERR1	Stop bit not found
	TAPE ERR2	Wrong data or program memory assignment
	TAPE ERR4	Program memory or file register capacity does not match controller capacity
	TAPE ERR5	Unmatched verification
	TAPE ERR6	Check code error
	TAPE ERR 7	Transmission error
Memory clear	MEM CL	Memory clear operation
	MEM CL OK	All memory clear successful
	MEM CL NG1	Program memory error
	MEM CL NG2	Data memory error
	MEM CL NG3	Program and data memory errors
	MEM CL NG4	System memory error
	FILE MEM CL	File memory clear
	FILE MEM CL OK	File memory clear OK
	FILE MEM CL NG	File memory error
	SYS MEM CL	System memory initialization
	SYS MEM CL OK	System memory initialization complete
	SYS MEM CL NG	System memory initialization error
Parity check	IN CHECK	Parity check in progress
	CHECKED	Parity check complete
	PARITY ERROR	Parity error exists
Program check	IN CHECK	Program check in progress
	CHECKED	Program check complete
	STACK OVER	Abuse of STR NOT instruction
	STACK UNDER	Lack of STR NOT instruction or abuse of AND STR or OR STR instruction
	STACK EXIST	Data remains in the stack after F-40
	MCR ERROR	F-31 not preceded by F-30 instruction
	*MCS EXIST	F-30 not reset before F-40 instruction
	JCS EXIST	F-41 not reset before F-40 instruction
	JCS ERROR	F-41 is nested within another F-41
	JCR ERROR	F-42 is used without preceding F-41
	DOUBLE OUT	OUT instruction used more than once
	DOUBLE NUMBER	TMR, CNT, MD number used more than once
	NO END ERROR	No F-40 instruction in program

Display Message List (Cont)

ITEM	MESSAGE		MEANING
Basic Instruction	STR		Start of the logical line, or store the result of the logical operation
	STR NOT		Use of a b-type contact in the above
	AND		AND
	AND NOT		AND NOT
	OR		OR
	OR NOT		OR NOT
	AND STR		AND STR with the intermediate result
	OR STR		OR STR with the intermediate result
	OUT		External or internal output
	TMR		Internal timer
	CNT		Internal counter
	MD		Maintenance display
Application Instruction	F-00	XFER	One byte transfer between data registers
	F-01	BCD	Transfer BCD constant
	F-02	XCHG	Data exchange between two registers
	F-03	→BIN	BCD to binary conversion
	F-04	→BCD	Binary to BCD conversion
	F-05	DMPX	Demultiplex
	F-06	MXP	Multiplex
	F-07	DCML	Transfer of decimal constant
	F-08	OCT	Transfer of octal constant
	F-09	INV	Transfer of inverted data
	F-10	ADD	Addition between registers
	Fc10	ADD	Addition of register and BCD constant
	F-11	SUB	Subtraction between registers
	Fc11	SUB	Subtraction of register and BCD constant
	F-12	CMP	Comparison between registers
	Fc12	CMP	Comparison of register and octal constant
	F-13	AND	AND between registers
	Fc13	AND	And of register and octal constant
	F-14	OR	OR between registers
	Fc14	OR	OR of register and octal constant
	F-15	MUL	Multiplication between registers
	Fc15	MUL	Multiply register with BCD constant
	F-16	DIV	Division between registers
	Fc16	DIV	Division of register and BCD constant
	F-17	XNR	XNR between registers
	Fc17	XNR	XNR between register and octal constant
	F-18	XOR	XOR between registers
	Fc18	XOR	XOR between register and octal constant
	F-30	MCS	Sets master control active
	F-31	MCR	Deactivates MCS
	F-40	END	Last line in programs
	F-41	JCS	Sets jump control active
	F-42	JCR	Deactivates JCS

Display Message List (Cont)

ITEM	MESSAGE		MEANING
Application Instruction	F-43		Bit inversion
	F-44	-|↑|-	Transitional contact-low to high = ON
	F-45	-|↓|-	Transitional contact-high to low = ON
	F-50	4→16	4 to 16 decode
	F-51	16→4	16 to 4 encode
	F-52	7 SEG	Decode for 7 segment display
	F-53	→BIN	BCD to binary conversion (expanded)
	F-54	→BCD	Binary to BCD conversion (expanded)
	F-55	SWAP	Swaps high order bits with low order
	F-60	SFR	Bidirectional shift register
	F-61	ASFR	Asynchronous shift register
	F-62	U/DC	BCD up down counter
	F-63	INC	Incremental counter
	F-64	DEC	Decremental counter
	F-70	FILE	n-byte batch transfer
	F-71	CONS	Octal constant batch transfer
	F-72	DMPX	Demultiplexing file register
	F-73	MPX	Multiplexing file register
	NOP		No operation

Appendix C

Data Memory

TYPE	CAPACITY	RELAY NUMBER (bit address)	BYTE ADDRESS *	ABSOLUTE ADDRESS	STATE AFTER POWER FAILURE
Input/Output relays	512 points (64 bytes)	0000 to 0777	].000~].077	000 to 077	Cleared (note 2)
Link relays	512 points (64 bytes)	2000 to 2777	].200~].277	100 to 177	Cleared (note 2)
Internal relays	512 points (64 bytes)	4000 to 4777	].400~].477	200 to 277	Cleared (note 2)
Latched relays	256 points (32 bytes)	7000 to 7377	].700~].737	300 to 337	Retained (note 2)
TMR, CNT, MD	128 points (current value 256 bytes)		b.000 to b.377		TMR clears CNT and MD retained (note 3)
Register	256 bytes		9000~9377		Retained
File Register	2048 bytes maximum		30000~33777 (note 4)		Retained

Note 1: All data memory addresses are octal. The 9000 series registers are activated by programming the Ever-ON contact with the preset value change coil in the latched special area.

Note 2: The latched relay area retains its last state before a power failure. It is possible to change the size of the area by using system memory #200. Areas not assigned as 'latched' will be cleared when power is restored.

Note 3: TMR will retain its state if system memory #201 is set.

Note 4: Capacity of the file register is determined by system memory #205.

I/O Relay	Area interfaced to input unit	ON/OFF state of the input unit is read into this area during I/O processing and will be retained for one scan. Used for input information (contacts, data) by the program
	Area interfaced to output unit	Operational result is stored in this area by user program in terms of coil/destination. ON/OFF state is transferred to the output unit of the I/O unit during I/O processing. Operational result can be used for contact and source in the program.
	Area that is not interfaced	Operational result is written into this area by the user's program. Can be used for internal and remote I/O relays.
Link Relays	Area used for data link receiving	In processing a request from data link module, data received from other PC's is stored here. Used for input information (contact, data) by the program.
	Area used for data link transmitting	Operational result is stored in this area by the user program in terms of coil and destination. In processing a request from data link modules, data in this area is transferred to the data link module buffer, for transfer data to other PC's. Operational result can be used for contact and source in the program.
	Area used for remote I/O output	Operational result is stored in this area by user's program. In processing a request from the remote I/O module, the data in this area is transferred to the remote I/O module memory buffer, to be transmitted to the remote I/O slave output unit. Operational result can be contact or source in program
	Area not used	Can be internal relay

Internal Relay	Area used for remote I/O input	Same as the link relay used for the remote I/O input.
	Area used for remote I/O output	Same as the link relay used for remote I/O output.
	Area not used for remote I/O	Operational result is stored in this area by user program in terms of coil/destination. Used for temporary storage of the operational result not needed for external output. Operational result can be used for contact or source.
Latched Relay	Area used for remote I/O input	Same as the link relay used for remote I/O input.
	Area used for remote I/O output	Same as the link relay used for remote I/O output.
	Area not used for the remote I/O	Operational result is stored in this area by the user program in terms of coil and destination. Used for temporary storage of the operational result which is not an external output Operational result can be used as contact and source in the program. Used to store contact and data information in case of a power failure.
	Special Area (7340 to 7377)	Used for error code and flag storage. It can not be used for coil/destination in the program, but can be used as contact and source.

TMR, CNT, MD	When used for TMR	When the current value is '0' the timer contact is ON. TMR contact can be used as many times as required. Current value can be used as source in the program
	When used for CNT	When the current value is '0' the CNT contact is ON. CNT contact can be used as many times as required. Current value can be used as the source in a program.
	When used for MD	The MD information is written in the current value area when the output direction is ON. MD information in the current value area can be used as a source in the program.
	Area not used	Area b.xxx can be used as a register.
Register		Operational result is stored in register by the program in terms of destination. Operational result is used as source in the program. It is possible in the preset value change mode to write data from the programmer.
File Register		Used when more than 256 bytes of register area (9000~9377) are required. Read and write is possible by the user program using the F-05, F-06, F-70, F-71, F-72, or F-73 instructions. It is possible in the preset value change mode to write data from the programmer.

Note 1: The area interfaced with the input unit can retain the ON/OFF state sensed during I/O processing until the I/O processing of the next cycle. Because the area is used for coil/destination by the program, the data memory may be revised depending on the operational result in that scan.

Note 2: The register into which the operational result is stored is the destination. The register into which the data before the operation is stored is the source.

Note 3: For detail of the data link module refer to the appropriate instruction manual.

Note 4: For detail of the latched relay special area refer to the latched relay special area section of this manual.

Note 5: For detail on the file register refer to program memory and file register.

Appendix D – Bit Manipulation for Basic Instruction Set

Controller Configuration

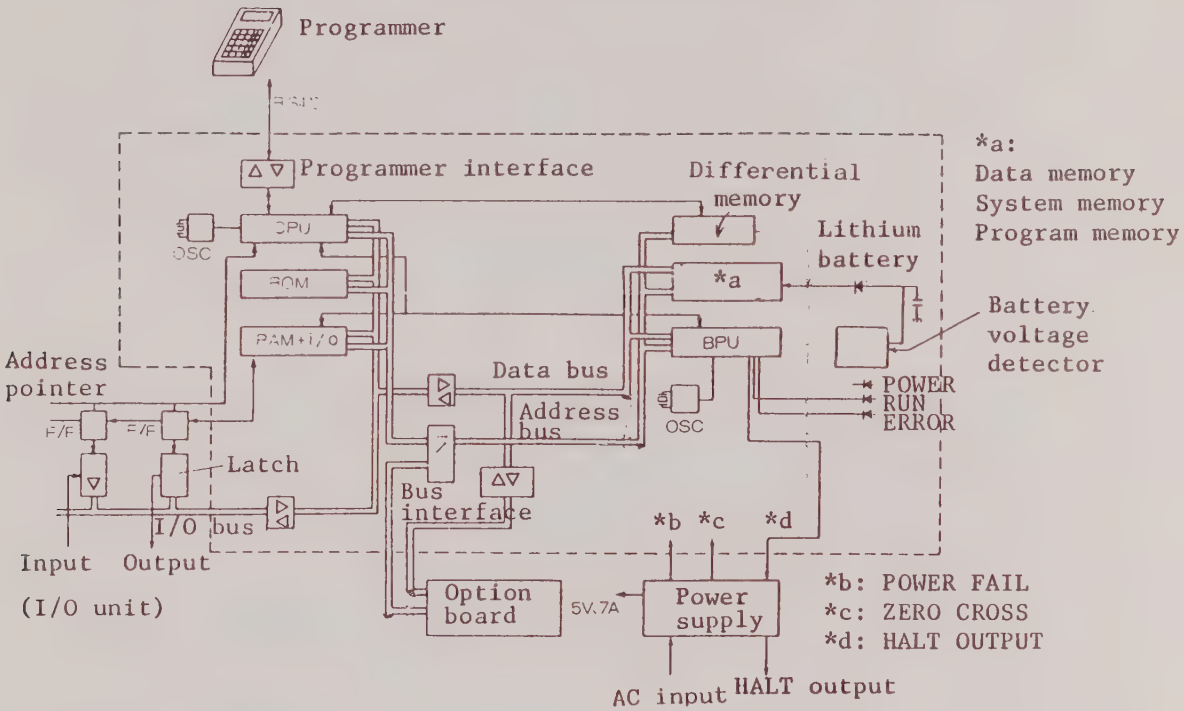


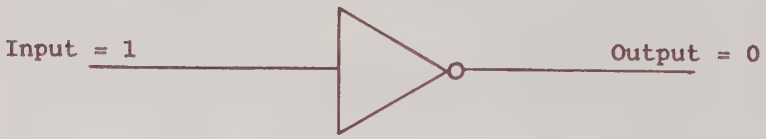
Fig. 63, Controller Block Diagram

Block Diagram Description

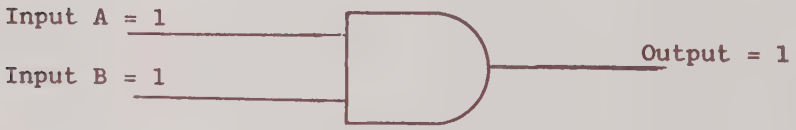
ITEM	DESCRIPTION
CPU	Controls the entire PC 0185 and has these functions: 1. Data transfer with the I/O card 2. Data communication with the I/O units 3. Program transfer to user memory 4. BPU control (for basic instruction set) 5. Sequence operation
BPU	A custom IC that performs the basic instruction set
I/O	Port that exchanges data between CPU and I/O devices
Program memory	Acts as the sequence for operation and file storage
Data memory	Storage for I/O, TMR, CNT, and registers
Special memory	Memory area used to detect rising and falling edges
System memory	Memory used for CPU functional designation, error code storage, and optional module designations

The PC 0185 has 12 basic programming instructions. The instructions, associated symbols, and a brief description of each command's function is shown on the following page. The PC 0185 offers the additional flexibility of combining N.O. and N.C. contacts with logical AND, OR, and NOT functions.

The NOT function causes the output to be the opposite of the input:



The AND function causes the output to go high (1), when, and only when, both of the inputs are high (1):



The OR function causes the output to go high (1) when either, or both, of the inputs is high (1):

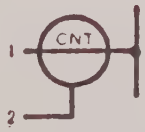
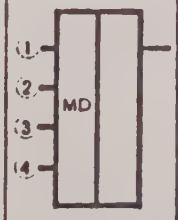
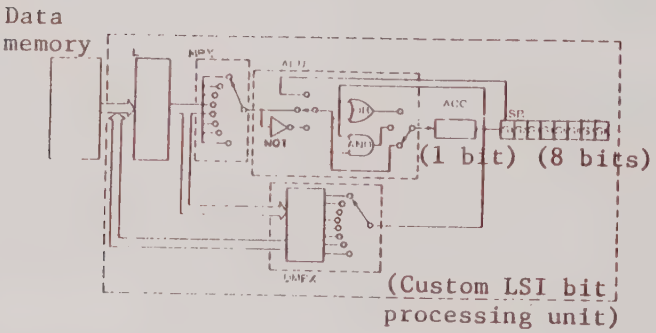
Instruction	Symbol	Step	Discription
STR		1	Programs a normally open contact at the beginning of logical circuit block. Store the result of the logic operation just bwfore this instruction is stored in the stack register.
STR NOT			Programs a normally closed contact at the beginning of logical circuit block. Store the result of the logic operation just bwfore this instruction is stored in the stack register.
AND			ANDs the result just before this instruction with the status of the next relay.
AND NOT			ANDs the result just before this instruction with the inverted status of the next relay.
OR			ORs the result just before this instruction with the status of the next relay.
OR NOT			ORs the result just before this instruction with the inverted status of the next relay.
AND STR			ANDs the result just before this instruction with the result stored in the stack register.
OR STR			ORs the result just before this instruction with the result stored in the stack register.
OUT		1	Outputs the result just before this instruction.
TMR		2	Internal timer Preset time : 0.1 - 199.9 sec
CNT			Internal counter Preset value : 1 - 1999 1: Count input 2: Reset input
MD			Maintenance display Preset value : 000 - 999 1: Operation command 2: Operation start condition 3: Operation end condition 4: Display command

Fig. 64, Basic Programming Instructions

Bit Manipulation

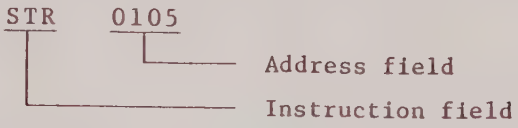
The PC 0185 processes all of the basic instructions (except TMR, CNT, and MD) by process called bit manipulation. The bit manipulation is carried out by an integrated circuit inside the PC 0185. When the bit manipulation is complete the instructions are executed by the CPU. The TMR, CNT, and MD instructions don't need this intermediate step, and are carried out by the CPU.



L	(latch)
MPX	(bit extraction)
ALU	(single bit logical unit)
ACC	(accumulator)
SR	(stack register)
DMPX	(bit insertion)

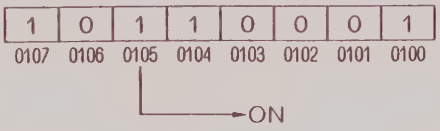
Fig. 65, Diagram of Bit Processing

1. L (Latch)
The bit processing procedure uses the instruction and address fields.

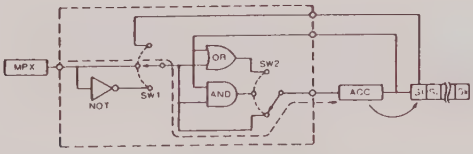


The address field represents the relay number of the data memory relay area (I/O relay, link relay, internal relay, or latched relay). When the state of the relay is read from the data memory, the byte containing that relay number is read and latched. In the previous example (STR 0105) the addresses between 0100 and 0107 (one byte) are read.

2. MPX (multiplexer)
The specific address bit is extracted. 0105 in this example.



3. ALU (arithmetic logic unit)
Logical operation takes place according to the contents if the instruction field.



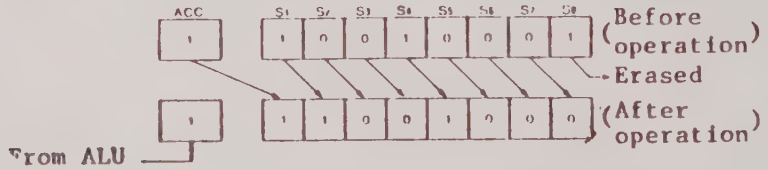
The figure above shows how the operation takes place for the STR instruction. Switches SW1 and SW2 are set according to the contents of the instruction field.

4. ACC (accumulator)
A flip-flop that stores the result of the ALU.

5. SR (stack register)
An eight bit bidirectional shift register that stores intermediate results. It is used in processing series/parallel circuits and application instructions with several inputs.

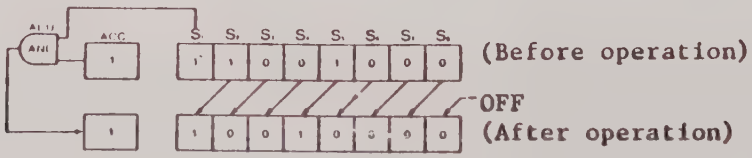
A. SR Action When Executing STR Instruction:
The ON/OFF state of the bit read from data memory is stored in the accumulator.

The ON/OFF data previously stored in the accumulator is placed in the shift register. The contents of each flip-flop is moved (shifted) one flip-flop to the right. The data stored in the last flip-flop is lost.



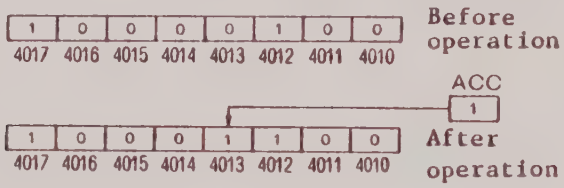
B. SR Action When Executing AND STR Instruction:
The ON/OFF data in S1 (first flip-flop of the shift register) is moved to the ALU, the ANDed with the contents of the ACC. The result is stored in the ACC.

The ON/OFF information in S1 is no longer needed and is erased. The contents of each flip-flop in the shift register is moved (shifted) one flip-flop to the left. The data stored in the last flip-flop is defaulted to 0.



6. DMPX (demultiplex)
With the OUT instruction is executed, the one bit multiplexed into the ALU and acted on, is transferred to the demultiplexed output. The entire eight bits is moved back to the data memory.

(In case the operational result is ON for OUT 4013)

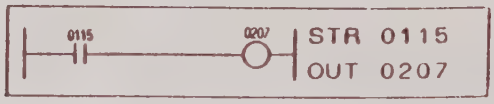


Description of the Basic Instructions

1. STR/ OUT

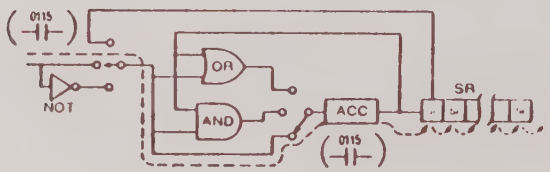
STR The ON/OFF state of the specified data memory location is stored in the accumulator. The previous contents of the accumulator are moved to the shift register.

OUT The contents of the accumulator are sent to the data memory area.



STR 0115

L..... The data memory byte 0110 to 0117 is read.
MPX..... 0115 is extracted from the latched address.
ALU, ACC, SR.. The MPX output is stored in the ACC. The previous contents of the accumulator are moved to S1.

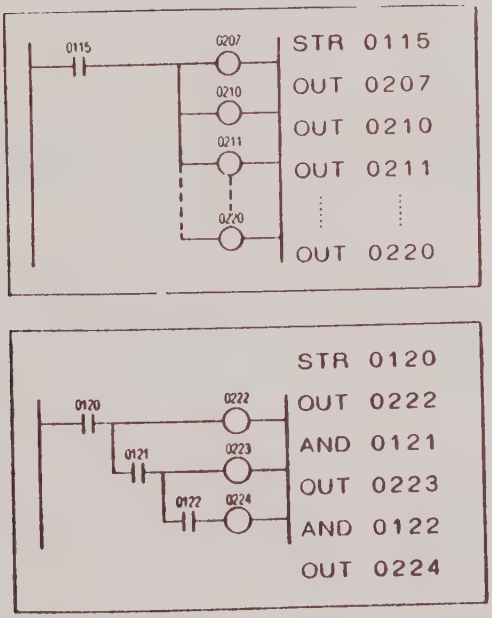


OUT 0207

L..... The byte 0200 to 0207 is read from data memory.

DMPX..... One bit (0207) in the latch (L) is replaced by the contents of the ACC. The entire byte (0200 to 0207) is sent back to the data memory area.

Since the contents of the ACC are not affected by the execution of the OUT instruction, the value stored in the ACC can be used repeatedly.

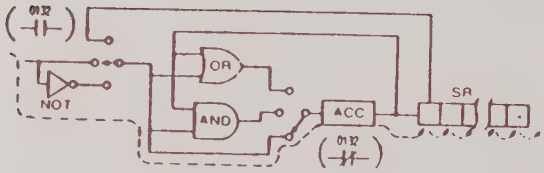


2. STR NOT

The contents of the specified data memory are inverted and stored in the ACC. The previous contents of the ACC are moved to the stack register (S1).

STR NOT

L..... The byte 0130 to 0137 is read from data memory.
MPX..... Bit 0132 is extracted from the latch.
ALU, ACC, SR.. The multiplexer output is inverted and written in the ACC. The previous contents of the ACC are moved to the stack register (S1).

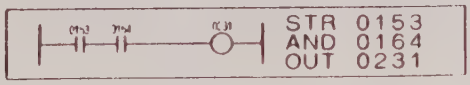


OUT 0210

The result of STR NOT is written to the data memory 0210, as described on page A-23.

3. AND

The contents of the specified data memory are ANDed with the contents of the accumulator. The result is stored in the accumulator.



STR 0153

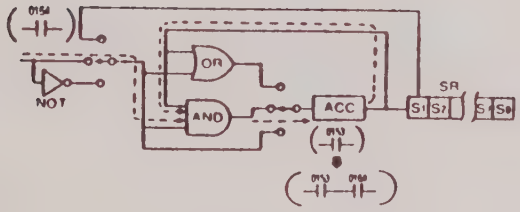
The contents of the data memory (0153) are stored in the accumulator, as shown on page A-22.

AND 0164

L..... The byte 0160 to 0167 is read from data memory.

MPX..... Bit 0164 is extracted from the latch.

ALU, ACC, SR.. The contents of the accumulator (0153) are ANDed with the MUX output (0164). The result is stored in the accumulator. The contents of the stack register are retained.



OUT 0231

The result of the AND operation is written to data memory 0231, as described on page A-23.

4. AND NOT

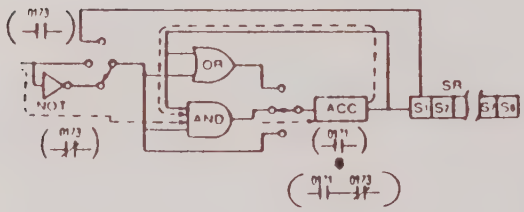
The contents of the specified data memory are inverted ANDed with the contents of the accumulator. The result is stored in the accumulator.

STR 0171

The contents of the data memory (0171) are stored in the accumulator, as shown on page A-22.

AND NOT 0173

L..... The byte 0170 to 0177 is read from data memory.
MPX..... Bit 0173 is extracted from the latch.
ALU, ACC, SR.. The inverted MUX output (0173) are ANDed with the contents of the accumulator (0171). The result is stored in the accumulator. The contents of the stack register are retained.

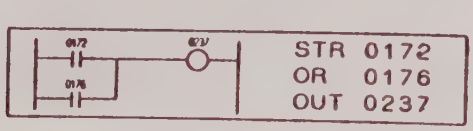


OUT 0235

The result of the AND NOT is written to data memory 0235, as described on page A-23.

5. OR

The contents of the specified data memory are ORed with the contents of the accumulator. The result is stored in the accumulator.

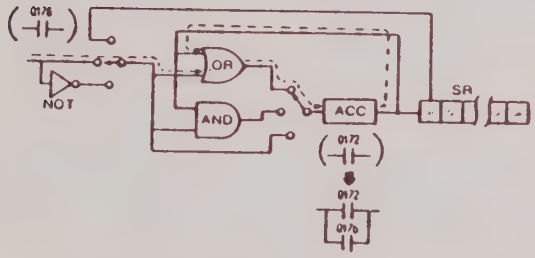


STR 0172

The contents of the data memory (0172) are stored in the accumulator as described on page A-22.

OR 0176

L.....	The byte 0170 to 0177 is read from the data memory.
MPX.....	Bit 0176 is extracted from the latch.
ALU, ACC, SR.....	The contents of the accumulator (0172) are ORed with the multiplexer output (0176) and the result is stored in the accumulator. The contents of the stack register are retained.



OUT 0237

The result of the OR is written to data memory 0237, as shown on page A-23.

6. OR NOT

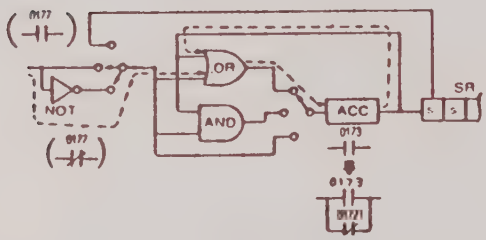
The contents of the specified data memory are ORed with the contents of the accumulator. The result is stored in the accumulator.

STR 0173

The contents of the data memory 0173 are stored in the accumulator as described on page A-22.

OR NOT 0177

L.....	The byte 0170 to 0177 is read from the data memory.
MPX.....	Bit 0177 is extracted from the latch.
ALU, ACC, SR.....	The inverted multiplexed output (0177) and the contents of the accumulator are ORed. The result is stored in the accumulator. The contents of the stack register are retained.

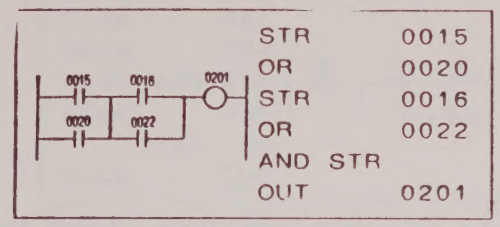


OUT 0240

The result of the OR NOT is written to data memory 0240, as shown on page A-23.

7. AND STR

The contents of the stack register is ANDed with the contents of the accumulator. The result is stored in the accumulator.



STR 0015

The contents of the data memory 0015 are stored in the accumulator as described on page A-22.

OR 0020

The result of the OR (0015, 0020) is stored in the accumulator. OR function described on page A-27.

STR 0016

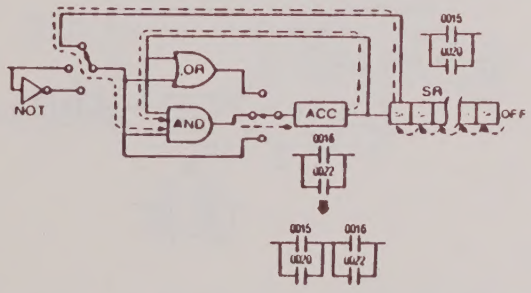
Data memory 0016 is stored in the accumulator and the previous contents (ORed 0015 with 0020) are moved to the stack register (S1).

OR 0020

The result of the OR (0016, 0022) is stored in the accumulator. OR function described on page A-27.

AND STR

ALU,ACC,SR.... The contents of S1 (ORed 0015, 0020) are ANDed with the contents of the accumulator (ORed 0016, 0022). The result is stored in the accumulator.

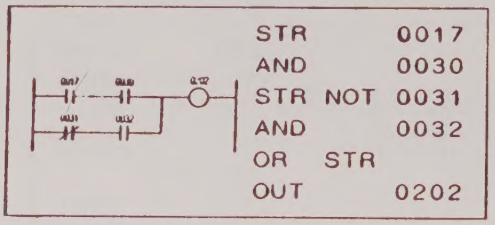


OUT 0201

The result of the AND STR is written to data memory 0201, as described on page A-23.

8. OR STR

The contents of the stack register is ORed with the contents of the accumulator. The result is stored in the accumulator.



STR 0017

The contents of the data memory 0017 are stored in the accumulator as described on page A-22.

AND 0030

The result of the AND (0017, 0030) is stored in the accumulator. AND function described on page A-25.

STR NOT 0031

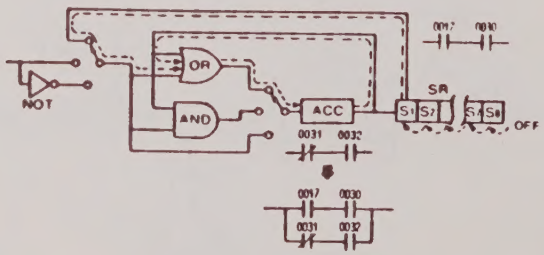
The accumulator contents are moved to the stack register (S1) and the contents of data memory 0031 are inverted and written to the accumulator. The STR NOT instruction is shown on page A-24.

AND 0032

The result of the AND (0031, 0032) is stored in the accumulator.

OR STR

ALU,ACC,SR.... The contents of S1 (ANDed 0017, 0030) are ORed with the contents of the accumulator (ANDed 0031, 0032). The result is stored in the accumulator.



OUT 0201

The result of the OR STR is written to data memeory 0202, as described on page A-23.

www.manualsplus.com

1033389

Gould Inc., Programmable Control Division
P.O. Box 3083, Andover, Mass 01810
(617) 475-4700 - Telex 443-0078
24 Hour Support Center (800) GOULD-PC

GOULD
Electronics